



## **Context Information Management (CIM); NGSI-LD Primer**

---

### *Disclaimer*

The present document has been produced and approved by the cross-cutting Context Information Management (CIM) ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG. It does not necessarily represent the views of the entire ETSI membership.

---

**Reference**

DGR/CIM-008-NGSI-LD-Primer

---

**Keywords**

API, interoperability, IoT, smart city

**ETSI**

650 Route des Lucioles  
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C  
Association à but non lucratif enregistrée à la  
Sous-Préfecture de Grasse (06) N° 7803/88

---

**Important notice**

The present document can be downloaded from:

<http://www.etsi.org/standards-search>

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format at [www.etsi.org/deliver](http://www.etsi.org/deliver).

Users of the present document should be aware that the document may be subject to revision or change of status.

Information on the current status of this and other ETSI documents is available at

<https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx>

If you find errors in the present document, please send your comment to one of the following services:

<https://portal.etsi.org/People/CommitteeSupportStaff.aspx>

---

**Copyright Notification**

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.

The content of the PDF version shall not be modified without the written authorization of ETSI.

The copyright and the foregoing restriction extend to reproduction in all media.

© ETSI 2020.

All rights reserved.

**DECT™**, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members.

**3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.

**oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners.

**GSM®** and the GSM logo are trademarks registered and owned by the GSM Association.

# Contents

Intellectual Property Rights .....	4
Foreword.....	4
Modal verbs terminology.....	4
Executive summary .....	4
Introduction .....	4
1 Scope .....	6
2 References .....	6
2.1 Normative references .....	6
2.2 Informative references.....	6
3 Definition of terms, symbols and abbreviations.....	6
3.1 Terms.....	6
3.2 Symbols.....	7
3.3 Abbreviations .....	8
4 Motivation and an Example Use Case .....	8
5 NGSI-LD Getting Started.....	9
5.1 Introduction .....	9
5.2 Architectural Assumptions .....	9
5.3 Creating NGSI-LD Entities and Properties .....	10
5.4 Retrieving NGSI-LD Entities and Properties .....	11
5.5 Overview .....	12
6 NGSI-LD Information Model .....	12
7 NGSI-LD Information Representation.....	15
8 NGSI-LD Information Provision.....	15
8.1 Overview NGSI-LD Information Provision .....	15
8.2 Creating/Deleting Entities .....	16
8.2.1 Create Entity Using Link Header.....	16
8.2.2 Delete Entity .....	17
8.3 Creating/Updating/Deleting Properties and Relationships .....	17
8.3.1 Appending an Entity Property/Relationship .....	17
8.3.2 Updating an Entity Property/Relationship .....	18
8.3.3 Partial Update of an Entity Property/Relationship.....	18
8.3.4 Deleting an Entity Property/Relationship .....	19
9 NGSI-LD Information Consumption .....	19
9.1 Overview NGSI-LD Information Consumption .....	19
9.2 Retrieving NGSI-LD Information .....	20
9.2.1 Retrieving NGSI-LD Entity Using Simplified Representation.....	20
9.3 NGSI-LD Query Language .....	20
9.3.1 Querying NGSI-LD Entities by Type .....	20
9.3.2 Querying NGSI-LD Entities by Type, Filtering by Property Value .....	21
9.3.3 Querying NGSI-LD Entities by Type, Filtering by Relationship Object.....	23
9.3.4 Querying NGSI-LD Entities by Type, Filtering by Meta Information .....	24
9.4 Geographical NGSI-LD Queries .....	24
10 NGSI-LD Information Subscriptions .....	25
10.1 Overview NGSI-LD Information Subscriptions.....	25
10.2 Creating Subscriptions .....	25
10.2.1 Change-based Subscriptions, Updates and Resulting Notifications .....	25
10.2.2 Time-based Subscriptions and Resulting Notifications .....	28
<b>Annex A: Change History .....</b>	<b>30</b>
History .....	31

---

## Intellectual Property Rights

### Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: "*Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards*", which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (<https://ipr.etsi.org/>).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

### Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

---

## Foreword

This Group Report (GR) has been produced by ETSI Industry Specification Group (ISG) cross-cutting Context Information Management (CIM).

---

## Modal verbs terminology

In the present document "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the [ETSI Drafting Rules](#) (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

---

## Executive summary

The present document (this "Primer") is intended to give developers an introduction on how the NGSI-LD API is used. The aim is to give developers, especially to those building applications and services on top of the NGSI-LD API, an easy start by explaining the NGSI-LD API based on typical examples. For illustration purposes a scenario is introduced, for which the information is modelled according to the NGSI-LD information model. Examples for providing information, i.e. creating, updating and deleting information, and for requesting information, i.e. synchronous queries as well as asynchronous subscribe/notify interactions, are given. The focus is on typical usage rather than on completeness of all features.

---

## Introduction

While ETSI GS CIM 009 [i.1] provides the complete specification of the NGSI-LD API, the present document, called "Primer", is intended to give users an introduction to the use of the NGSI-LD API. The idea is to take a simple scenario, i.e. a store that sells products to customers, for illustration purposes and show typical NGSI-LD API operation examples. The examples for information provision show how entities, properties and relationships can be created, updated, appended and deleted. The examples for information consumption show how entities can be synchronously queried, filtered according to property values or filtered according to geographical location using geographic queries. Finally change-based and time-based subscriptions are introduced and how these create asynchronous subscriptions depending on a change-based or time-based trigger.

Most of the present document (NGSI-LD Primer) was created with the support of the following European Union Horizon 2020 research projects: No. 732851 (FI-NEXT), No. 723156 (WISE-IoT), No. 732240 (SynchroniCity) and No. 731993 (AutoPilot), No. 814918 (Fed4IoT), No. 779852 (IoTcrawler), No. 731884 (IoF2020).

---

# 1 Scope

The present document provides an introduction, in particular for developers, on how the NGSI-LD API, defined in ETSI GS CIM 009 [i.1], is used. The focus is on typical use and is based on a small NGSI-LD information model example. More information about the NGSI-LD information model can be found in ETSI GR CIM 002 [i.2].

---

## 2 References

### 2.1 Normative references

Normative references are not applicable in the present document.

### 2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document, but they assist the user with regard to a particular subject area.

[i.1] ETSI GS CIM 009: "Context Information Management (CIM); NGSI-LD API".

[i.2] ETSI GR CIM 002: "Context Information Management (CIM); Use Cases (UC)".

[i.3] IETF RFC 3986: "Uniform Resource Identifier (URI): Generic Syntax".

NOTE: Available at <https://tools.ietf.org/html/rfc3986>.

[i.4] IEEE POSIX 1003.2™-1992: "IEEE Standard for Information Technology - Portable Operating System Interfaces (POSIX®) - Part 2: Shell and Utilities".

[i.5] IETF RFC 7946: "The GeoJSON Format".

NOTE: Available at <https://tools.ietf.org/html/rfc7946>.

---

## 3 Definition of terms, symbols and abbreviations

### 3.1 Terms

For the purposes of the present document, the terms given in ETSI GS CIM 009 [i.1] and the following apply:

NOTE: The letters "NGSI-LD" were added to most terms to confirm that they are distinct from other terms of similar/same name in use in other organizations, however, in the present document the letters "NGSI-LD" are generally omitted for brevity.

**NGSI-LD Attribute:** reference to either the name of an NGSI-LD Property or to the name of an NGSI-LD Relationship

**NGSI-LD Central Broker:** NGSI-LD Context Broker that only uses a local storage when serving NGSI-LD requests, without involving any external Context Sources

**NGSI-LD Context Broker:** architectural component that implements all the NGSI-LD interfaces

**NGSI-LD Context Consumer:** agent that uses the query and subscription functionality of NGSI-LD to retrieve context information

**NGSI-LD Context Producer:** agent that uses the NGSI-LD context provision and/or registration functionality to provide or announce the availability of its context information to an NGSI-LD Context Broker

**NGSI-LD Entity:** informational representative of something that is supposed to exist in the real world, physically or conceptually

NOTE: In the NGSI-LD API, any instance of such an entity is **uniquely identified by a URI**, and characterized by reference to one or more **NGSI-LD Entity Type(s)**. The API defined by the present document only allows associating one NGSI-LD Entity Type per NGSI-LD Entity. This restriction will be removed in future versions.

**NGSI-LD Entity Type:** categorization of an NGSI-LD Entity as belonging to a class of similar entities, or sharing a set of characteristic properties

NOTE: In the NGSI-LD API, an NGSI-LD Entity Type is **uniquely identified by a URI**.

EXAMPLE 1: "Vehicle" is an NGSI-LD Entity Type and is identified with a proper URI.

EXAMPLE 2: Bob's private car whose plate number is "ABCD1234" is an NGSI-LD Entity whose NGSI-LD Entity Type Name is "Vehicle".

**NGSI-LD Linked Entity:** NGSI-LD Entity referenced from another NGSI-LD Entity (the linking NGSI-LD Entity) via an NGSI-LD Relationship

**NGSI-LD Linking Entity:** NGSI-LD Entity which is the subject of a Relationship to another NGSI-LD Entity (the linked NGSI-LD Entity) or an external resource (identified by a URI)

**NGSI-LD Name:** short-hand string (term) that locally identifies an NGSI-LD Entity Type, Property Type or Relationship Type and which can be mapped to a URI which serves as a fully qualified identifier

EXAMPLE: The sentence "Bob's vehicle's speed is 40 km/h" can be represented by an NGSI-LD Property, whose Name is "speed", and which characterizes an NGSI-LD Entity, which NGSI-LD Type Name is "Vehicle". Such a name can be expanded to a fully qualified name in the form of a URI, for instance "http://example.org/Vehicle" or "http://example.org/speed".

**NGSI-LD Property:** description instance which associates a main characteristic, i.e. an **NGSI-LD Value**, to either an NGSI-LD Entity, an NGSI-LD Relationship or another NGSI-LD Property and that uses the special *hasValue* property to define its target value

**NGSI-LD Relationship:** description of a directed link between a subject which is either an NGSI-LD Entity, an NGSI-LD Property, or another NGSI-LD Relationship on one hand, and an object, which is an NGSI-LD Entity, on the other hand, and which uses the special *hasObject* property to define its target object

EXAMPLE: An NGSI-LD Entity of type (Type Name) "Vehicle" (when parked) can be the subject of an NGSI-LD Relationship which object is an NGSI-LD Entity of type "Parking".

**NGSI-LD Value:** JSON value (i.e. a string, a number, true or false, an object, an array), or a JSON-LD typed value (i.e. a string as the lexical form of the value together with a type, defined by an XSD base type or more generally an IRI), or a JSON-LD structured value (i.e. a set, a list, a language-tagged string)

EXAMPLE: Bob's private car 'speed' NGSI-LD Value is the number 100 (kilometres per hour).

## 3.2 Symbols

Void.

### 3.3 Abbreviations

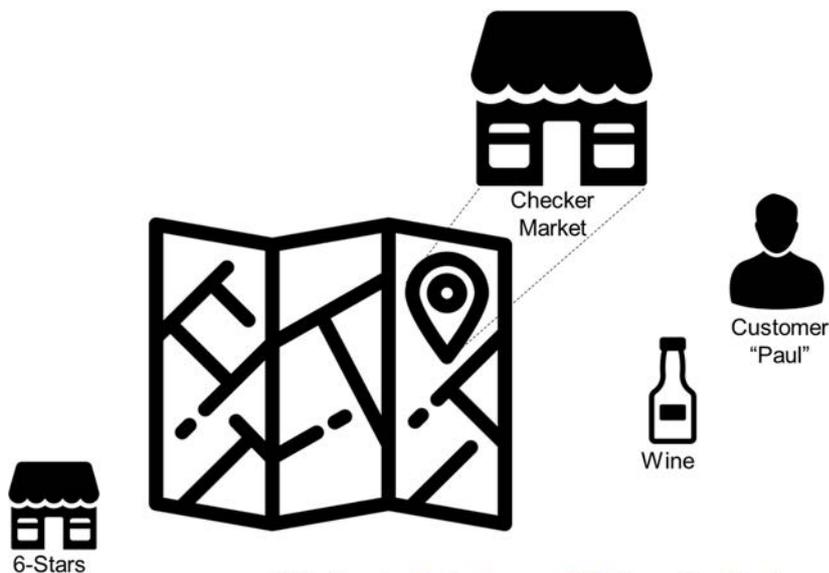
For the purposes of the present document, the abbreviations given in ETSI GS CIM 009 [i.1] and the following apply:

API	Application Programming Interface
HTTP	Hypertext Transfer Protocol
IETF	Internet Engineering Task Force
IoT	Internet of Things
IRI	Internationalized Resource Identifier
ISG	Industry Specification Group
JSON	JavaScript Object Notation
JSON-LD	JSON Linked Data
NGSI	Next Generation Service Interfaces
NGSI-LD	NGSI Linked Data
POSIX	Portable Operating System Interface
RFC	Request For Comments
URI	Uniform Resource Identifier
URL	Universal Resource Locator

## 4 Motivation and an Example Use Case

The concept of entity is at the core of the NGSI-LD model. Entities represent physical or conceptual objects existing in the real world. Entities can have properties describing aspects of the object the entity stands for and relationships to other entities. What kind of properties entities and relationships have is determined by the entity type?

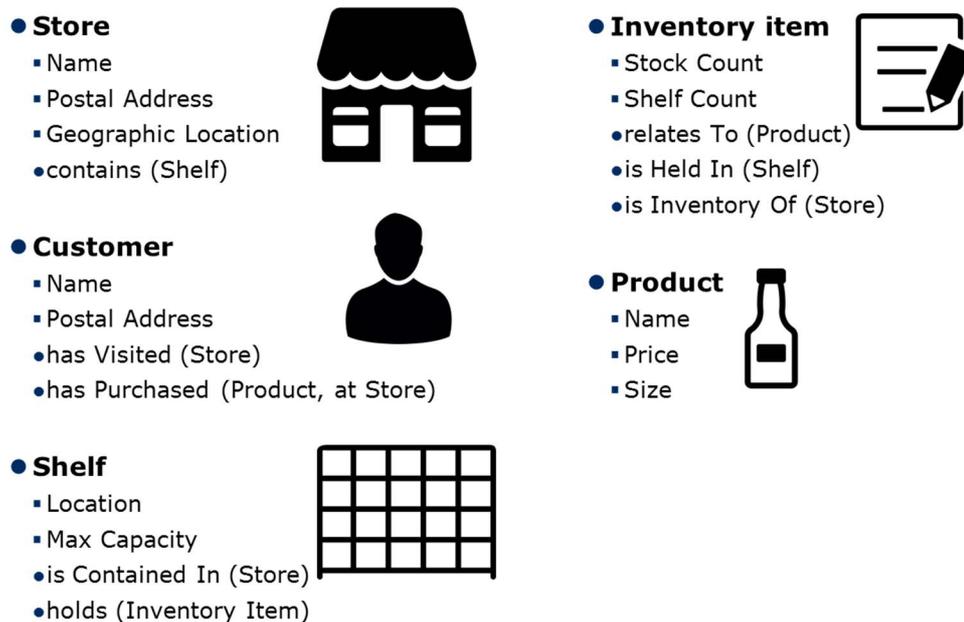
As example use case, the present document is using a system for managing context information related to grocery stores as depicted in Figure 4.1. It shows two instances of grocery store "6-Stars" and "Checker Market" with its location depicted on a map, a product "Wine" and a customer "Paul".



All clipart is under [Creative Commons BY 4.0](https://www.svgrepo.com) Licence from <https://www.svgrepo.com>

**Figure 4.1: Grocery store use case example**

The entity types used in the example are Store, Customer, Shelf, Inventory item and Product. Figure 4.2 shows the entity types together with the properties and relationships that instances of the respective entity type can have. As a convention for this example, properties are defined as nouns, whereas relationships are defined as verbs. (The use of this convention is not a requirement of NGSI-LD).



**Figure 4.2: Entity types, relationships and properties of use case**

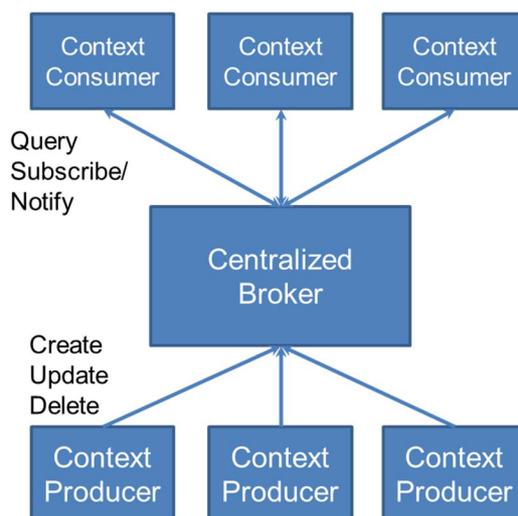
## 5 NGSI-LD Getting Started

### 5.1 Introduction

The purpose of clause 5 is to give a first introduction to the NGSI-LD representation and API operations using the HTTP binding. The examples can be used in an HTTP client (e.g. Postman or curl), targeting an NGSI-LD implementation, i.e. an NGSI-LD Context Broker.

### 5.2 Architectural Assumptions

NGSI-LD defines an API together with an underlying information model. It does not define a specific system architecture, but instead it is envisioned that the NGSI-LD API can be used in different architectural settings and the architectural assumptions of the API are kept to a minimum. For the following examples, the present document is using the simplest architectural setup, i.e. a centralized NGSI-LD Context Broker that stores all information. The resulting architecture is depicted in Figure 5.1 and requires a certain subset of NGSI-LD operations that are introduced in clauses 5, 6, 7, 8, 9 and 10.



**Figure 5.1: Basic architectural assumptions**

The roles in this setup are Context Producers, Context Consumers and a Centralized Broker. The assumption is that the Centralized Broker stores all information. Context Producers manage information, i.e. create, update and delete it, whereas Context Consumers synchronously query information or subscribe to information to be asynchronously notified. In the following, the assumption is that the Centralized Broker exposes the NGSI-LD API on `localhost:9090`. Context Producers and Context Consumers are roles. The same software program can have both roles at the same time, i.e. manage and request information.

It is planned to introduce more advanced architectural options in a future version of this Primer.

## 5.3 Creating NGSI-LD Entities and Properties

As entities are at the core of NGSI-LD, the following HTTP request creates a store entity with the id `urn:ngsi-ld:Store:001` of type `Store` (mapped to `https://uri.etsi.org/ngsi-ld/primer/Store`) in `@context`, the properties `address` and `storeName` and the GeoProperty `location` (all mapped to the respective URI concepts in `@context`).

```

POST /ngsi-ld/v1/entities/ HTTP/1.1
Host: localhost:9090
Content-Type: application/ld+json

{
  "@context": [
    {
      "Store": "https://uri.etsi.org/ngsi-ld/primer/Store",
      "address": "https://uri.etsi.org/ngsi-ld/primer/address",
      "storeName": "https://uri.etsi.org/ngsi-ld/primer/storeName",
      "streetAddress": "https://uri.etsi.org/ngsi-ld/primer/streetAddress",
      "addressRegion": "https://uri.etsi.org/ngsi-ld/primer/addressRegion",
      "addressLocality": "https://uri.etsi.org/ngsi-ld/primer/addressLocality",
      "postalCode": "https://uri.etsi.org/ngsi-ld/primer/postalCode"
    },
    "https://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context.jsonld"
  ],
  "id": "urn:ngsi-ld:Store:001",
  "type": "Store",
  "address": {
    "type": "Property",
    "value": {
      "streetAddress": "Main Street 65",
      "addressRegion": "Metropolis",
      "addressLocality": "Duckburg",
      "postalCode": "42000"
    }
  },
  "location": {
    "type": "GeoProperty",
    "value": {
      "type": "Point",

```

```

    "coordinates": [57.4874121, -20.2845607]
  },
  "storeName": {
    "type": "Property",
    "value": "Checker Market"
  }
}

```

**Figure 5.2: NGSi-LD entity creation**

If the creation was successful, the response in Figure 5.3 with HTTP return code 201 Created is returned.

```

HTTP/1.1 201 Created
Date: Wed, 03 Apr 2019 15:08:33 GMT
location: /ngsi-ld/v1/entities/urn:ngsi-ld:Store:001

```

**Figure 5.3: NGSi-LD entity creation result**

NGSi-LD defines the three special properties `location`, `observationSpace` and `operationSpace` as `GeoProperty`. A `GeoProperty` encodes a geographical location in GeoJSON format. A `GeoProperty` can be used for the scope of geographic queries, whereby the API specification requires that such queries will only return results based on that scope.

## 5.4 Retrieving NGSi-LD Entities and Properties

Now that the store entity with the id `urn:ngsi-ld:Store:001` has been created, it can be retrieved. The request is shown in Figure 5.4 and the successful result in Figure 5.5. Note that since no `@context` was provided in the request, only the core context is returned, whereas all user-defined aspects are returned with fully qualified names. How to provide the `@context` in requests without body is presented in clause 8.

```

GET /ngsi-ld/v1/entities/urn:ngsi-ld:Store:001 HTTP/1.1
Host: localhost:9090
Accept: application/ld+json

```

**Figure 5.4: NGSi-LD entity retrieval**

```

HTTP/1.1 200 OK
Date: Wed, 03 Apr 2019 15:50:09 GMTContent-Type: application/ld+json

{
  "id" : "urn:ngsi-ld:Store:001",
  "type" : "https://uri.etsi.org/ngsi-ld/primer/Store",
  "location" : {
    "type" : "GeoProperty",
    "value" : {
      "type" : "Point",
      "coordinates" : [ 57.4874121, -20.2845607 ]
    }
  },
  "https://uri.etsi.org/ngsi-ld/primer/address" : {
    "type" : "Property",
    "value" : {
      "https://uri.etsi.org/ngsi-ld/primer/addressLocality" : "Duck Village",
      "https://uri.etsi.org/ngsi-ld/primer/addressRegion" : "Metropolis",
      "https://uri.etsi.org/ngsi-ld/primer" : "42000",
      "https://uri.etsi.org/ngsi-ld/primer" : "Main Street 65"
    }
  },
  "https://uri.etsi.org/ngsi-ld/primer/storeName" : {
    "type" : "Property",
    "value" : "Checker Market"
  },
  "@context" : [ "https://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context.jsonld" ]
}

```

**Figure 5.5: NGSi-LD entity retrieval result**

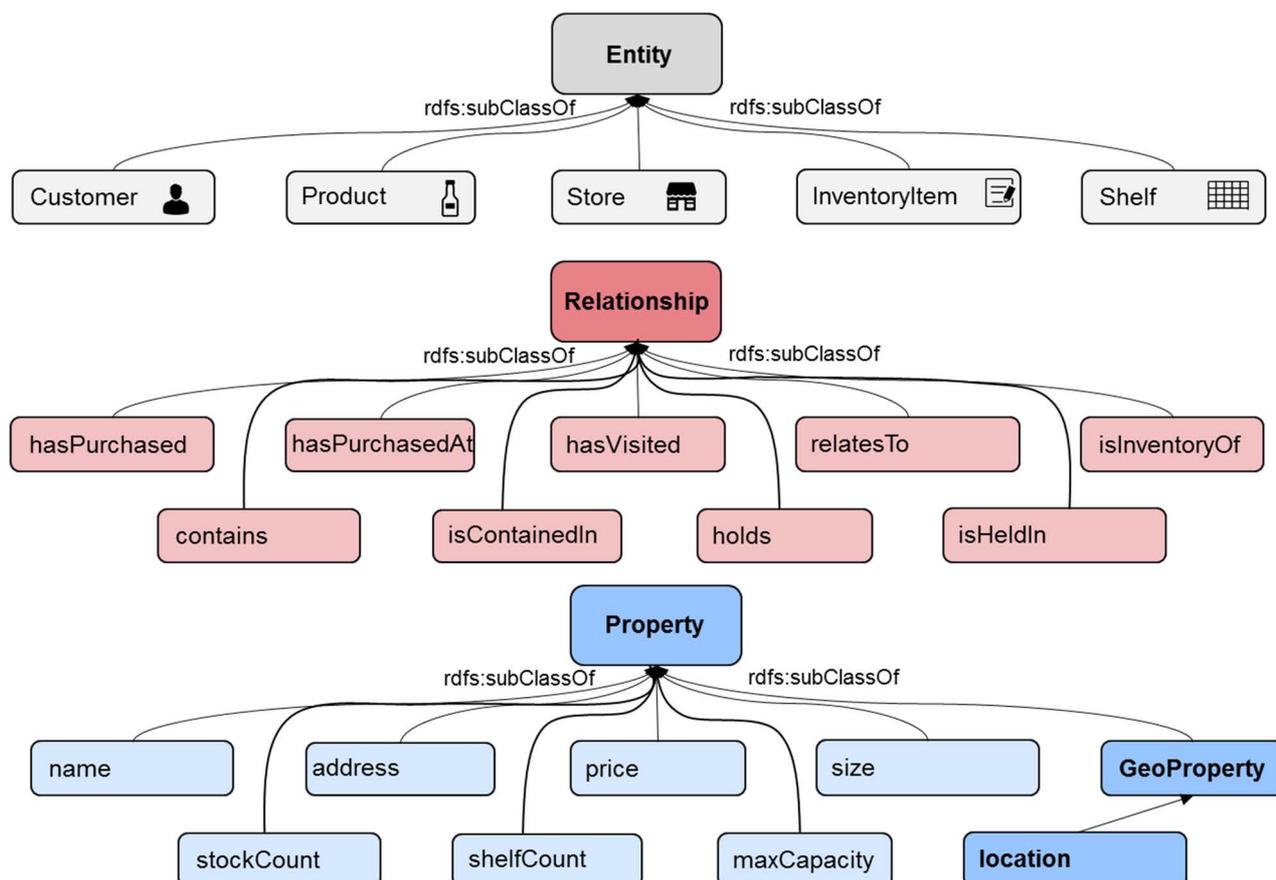
## 5.5 Overview

Clause 6 explains the NGSI-LD Information Model in more detail and clause 7 focuses on the NGSI-LD representation in JSON-LD and how it is used in the specified HTTP binding. Clause 8 gives typical examples of how NGSI-LD information is managed, i.e. created, appended, updated and deleted. Clause 9 describes how to synchronously retrieve and query NGSI-LD information and clause 10 gives examples for subscribing to NGSI-LD information and asynchronously retrieve notifications.

Developers eager to experiment with NGSI-LD operations can also decide to immediately jump to clause 8 and clause 9, and only go back to clause 6 and clause 7 in case there are open questions.

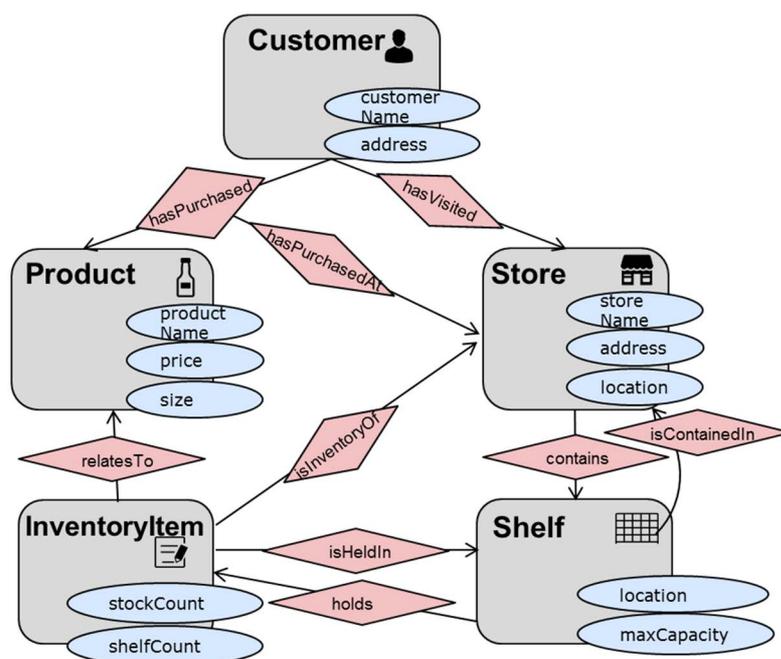
## 6 NGSI-LD Information Model

As already introduced, the core underlying concepts of the NGSI-LD Information Model, also referred to as the NGSI-LD Meta-Model, are *Entity*, *Relationship* and *Property*. Logically, the types of entities, relationships and properties are modelled as subclasses of the core concepts *Entity*, *Relationship* and *Property* respectively. For our example use case, the types of entities, relationships and properties are shown in Figure 6.1. Note that the concepts defined in the NGSI-LD specification are shown in bold. This includes the special property *location*, which is defined as a *GeoProperty* in the specification.



**Figure 6.1: Entities, properties and relationships of the use case**

The subclass relationships can be explicitly modelled in an ontology, but for the purpose of using the NGSI-LD API, they can also be implicitly assumed based on how they are used, i.e. as entity types or relationship and property names. The important aspect for the use in the NGSI-LD API is that the respective concepts are explicitly defined as URIs in the `@context`.



All clipart is under [Creative Commons BY 4.0](https://www.svgrepo.com) Licence from <https://www.svgrepo.com>

**Figure 6.2: Visualization of entity types, properties and relationships of the use case**

Figure 6.2 visualizes the entity types as rounded rectangles with the respective properties inside, whereas the relationships are shown as red coloured diamonds on directed arrows.

Properties have an entity of a certain type or another property or relationship as their domain and have a value as their range. The domain defines of what type the respective entities can be for which the property appears, e.g. a property "price" makes sense for a product, but not for a customer. The range indicates what type of value they can have, e.g. a "stockCount" for example would need to be of type integer. In Figure 6.2 the properties are shown together with the entity type that can be their domain. Domain and range of a certain property can be defined in an ontology, but this is not required for use in the NGSI-LD API.

Relationships in NGSI-LD are directed. Again, their domain and range indicate what types of source and target entities are valid for the relationship. For example, the "hasPurchased" relationship is only valid for customer entities as domain and product entities as range. In Figure 6.2 possible directed relationships are visualized as red coloured diamonds on directed arrows. In order to have inverse relationships, these need to be explicitly modelled, e.g. as the "contains" and "isContainedIn" relationships between Store and Shelf/Shelf and Store respectively.

For the JSON-LD based representation of NGSI-LD, it is important to have the mapping of the short terms used in the information representation to the URIs uniquely identifying the entity types, relationships and properties mapped in the @context. Figure 6.3 shows the complete "@context" for all the entities, relationships and properties introduced in our example use case. Note that the GeoProperty "location" is not defined here as it is already part of the NGSI-LD core context.

The NGSI-LD core context is stored in file `ngsi-ld-core-context.jsonld`, which can be retrieved from <https://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context.jsonld>. As the NGSI-LD terms take precedence over any user-defined terms, the reference to the NGSI-LD core context is required by the API specification to always be the last element of any NGSI-LD compatible @context. In case it is not provided in a request, NGSI-LD systems will implicitly add it as the last element.

```
"@context": [{
  "Customer": "https://uri.etsi.org/ngsi-ld/primer/Customer",
  "Product": "https://uri.etsi.org/ngsi-ld/primer/Product",
  "InventoryItem": "https://uri.etsi.org/ngsi-ld/primer/InventoryItem",
  "Store": "https://uri.etsi.org/ngsi-ld/primer/Store",
  "Shelf": "https://uri.etsi.org/ngsi-ld/primer/Shelf",
  "hasPurchased": "https://uri.etsi.org/ngsi-ld/primer/hasPurchased",
  "hasPurchasedAt": "https://uri.etsi.org/ngsi-ld/primer/hasPurchasedAt",
  "hasVisited": "https://uri.etsi.org/ngsi-ld/primer/hasVisited",
  "relatesTo": "https://uri.etsi.org/ngsi-ld/primer/relatesTo",
  "isInventoryOf": "https://uri.etsi.org/ngsi-ld/primer/isInventoryOf",
```

```

    "contains": "https://uri.etsi.org/ngsi-ld/primer/contains",
    "isContainedIn": "https://uri.etsi.org/ngsi-ld/primer/isContainedIn",
    "holds": "https://uri.etsi.org/ngsi-ld/primer/holds",
    "isHeldIn": "https://uri.etsi.org/ngsi-ld/primer/isHeldIn",
    "productName": "https://uri.etsi.org/ngsi-ld/primer/productName",
    "customerName": "https://uri.etsi.org/ngsi-ld/primer/customerName",
    "address": "https://uri.etsi.org/ngsi-ld/primer/address",
    "storeName": "https://uri.etsi.org/ngsi-ld/primer/storeName",
    "streetAddress": "https://uri.etsi.org/ngsi-ld/primer/streetAddress",
    "addressRegion": "https://uri.etsi.org/ngsi-ld/primer/addressRegion",
    "addressLocality": "https://uri.etsi.org/ngsi-ld/primer/addressLocality",
    "postalCode": "https://uri.etsi.org/ngsi-ld/primer/postalCode",
    "price": "https://uri.etsi.org/ngsi-ld/primer/price",
    "size": "https://uri.etsi.org/ngsi-ld/primer/size",
    "stockCount": "https://uri.etsi.org/ngsi-ld/primer/stockCount",
    "shelfCount": "https://uri.etsi.org/ngsi-ld/primer/shelfCount",
    "maxCapacity": "https://uri.etsi.org/ngsi-ld/primer/maxCapacity"
  } ,
  "https://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context.jsonld"
]

```

**Figure 6.3: @context of example use case**

NGSI-LD enables providing meta information to properties and relationships. This is done in the form of properties of properties, properties of relationships, relationships of properties and relationships of relationships. An example of a property of a property could be the timestamp when the property was created or modified, or quality information. A relationship of a property could refer to the entity representing the source of information. The same applies for relationships.

```

"storeName": {
  "type": "Property",
  "value": "6-Stars",
  "createdAt": "2019-08-09T18:08:02.669000Z",
  "providedBy": {
    "type": "Relationship",
    "object": "urn:ngsi-ld:Person:123"
  }
}

```

**Figure 6.4: Example of a Relationship of Property**

An example is shown for the property `storeName` in Figure 6.4. A relationship to the entity representing the person who provided the `storeName` property is given. In general properties and relationships of properties are modelled as regular properties or relationships, i.e. they are JSON objects with the respective type. An exception are specific timestamps that are modelled as temporal properties, i.e. `createdAt`, `modifiedAt` and `observedAt`, which are all defined in NGSI-LD's core `@context`. These, and only these special timestamps, are represented directly by their value, not as a JSON object and thus they cannot have any further properties or relationships.

As already mentioned in clause 5.3, a further special type of properties are GeoProperties. They are provided with type `GeoProperty` instead of `Property` and the API specification requires that their value be provided in GeoJSON [i.5]. Figure 6.5 shows an example of the `location` GeoProperty, which is a special GeoProperty defined by NGSI-LD and is part of the core `@context`. It is also possible to have user-defined GeoProperties.

```

"location": {
  "type": "GeoProperty",
  "value": {
    "type": "Point",
    "coordinates": [57.5522, -20.3484]
  }
}

```

**Figure 6.5: Example of a GeoProperty**

Only GeoProperties can be used to filter entities according to their geographic location as is explained in clause 9.4.

## 7 NGSI-LD Information Representation

NGSI-LD is represented in JSON-LD, which stands for JavaScript Object Notation for Linked Data. JSON-LD is an ordinary JSON document, but it contains specific reserved names that only JSON-LD readers can interpret (like `@id`, `@type` and `@context`). NGSI-LD maps `id` to `@id` and `type` to `@type`, so only `@context` may appear directly, i.e. outside value of `@context` in an NGSI-LD document.

As presented in clause 6, the most important aspect of JSON-LD is `@context` that maps short terms to unique URIs, identifying unique concepts for entity types, properties and relationships. Using the short terms in the core of the documents leads to a more readable and succinct JSON representation of the information itself.

In addition to this mapping, types can be specified in cases the data type required for a value is not a simple JSON type. Figure 7.1 shows an excerpt of the NGSI-LD core `@context` that defined the terms required in the NGSI-LD API specification itself – and can be retrieved through the URL <http://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context.jsonld>. For example, the API specification requires that the value of the `observedAt` temporal property be of type `DateTime`.

```

"@context": {
  "ngsi-ld": "https://uri.etsi.org/ngsi-ld/",
  "id": "@id",
  "type": "@type",
  "value": "https://uri.etsi.org/ngsi-ld/hasValue",
  "object": {
    "@id": "https://uri.etsi.org/ngsi-ld/hasObject",
    "@type": "@id"
  },
  ...
  "observedAt": {
    "@id": "https://uri.etsi.org/ngsi-ld/observedAt",
    "@type": "DateTime"
  },
  ...

```

**Figure 7.1: Excerpt of NGSI-LD core `@context`**

In an HTTP binding, there are two general approaches for providing the `@context` of JSON-LD. Either the `@context` is provided as part of the JSON document in the HTTP body, e.g. as shown in the example in Figure 5.2, or a URL to the `@context` is provided as an HTTP Link header. Of course, the former way of providing an `@context` is only feasible if there is an HTTP body as in POST, PUT or PATCH requests.

## 8 NGSI-LD Information Provision

### 8.1 Overview NGSI-LD Information Provision

Clause 8 describes the NGSI-LD API operations for information provision, i.e. the operations that can be used to provide and manage entity information. Clause 8.2 describes the operations for creating and deleting complete entities. Clause 8.3 describes the operations for appending, updating and deleting attributes (properties and relationships). This includes the partial update of attribute information, i.e. the API specification requires that only the aspects of the fragment with the aspects of the attribute to be updated need to be provided.

## 8.2 Creating/Deleting Entities

### 8.2.1 Create Entity Using Link Header

Figure 8.1 shows the NGSI-LD request to create another Store entity with its attributes. In this case, the @context is provided as a Link in the HTTP header pointing to the URL `https://uri.etsi.org/ngsi-ld/primer/store-context.jsonld`. This means that the @context needed to expand the JSON-LD information in the body can be retrieved from this URL. This is reflected by the Content-Type `application/json`. The alternative is shown in Figure 5.3, where the @context is part of the JSON in the HTTP body. The Content-Type in that case is `application/ld+json`. For brevity, all further operations presented in clause 8 use the Link header for referring to the @context and thus the `application/json` content type, which in this case contains the complete @context of the example use case. Alternatively, the @context could be provided in the HTTP body using the content type `application/ld+json`, given that the HTTP operation does have a body.

```
POST /ngsi-ld/v1/entities/ HTTP/1.1
Host: localhost:9090
Link: <https://uri.etsi.org/ngsi-ld/primer/store-context.jsonld>;rel="http://www.w3.org/ns/json-ld#context";type="application/ld+json"
Content-Type: application/json

{
  "id": "urn:ngsi-ld:Store:002",
  "type": "Store",
  "address": {
    "type": "Property",
    "value": {
      "streetAddress": "Tiger Street 4",
      "addressRegion": "Metropolis",
      "addressLocality": "Cat City",
      "postalCode": "42420"
    }
  },
  "location": {
    "type": "GeoProperty",
    "value": {
      "type": "Point",
      "coordinates": [57.5522, -20.3484]
    }
  },
  "storeName": {
    "type": "Property",
    "value": "6-Stars"
  }
}
```

**Figure 8.1: NGSI-LD Store entity creation using link header**

If the creation was successful, the response in Figure 8.2 with HTTP return code 201 Created is returned.

```
HTTP/1.1 201 Created
Date: Wed, 04 Apr 2019 11:42:15 GMT
location: /ngsi-ld/v1/entities/urn:ngsi-ld:Store:002
```

**Figure 8.2: NGSI-LD Store entity creation result**

For the example use case, further entities are needed. Figure 8.3 gives an example for creating a Shelf entity and Figure 8.4 shows the successful result.

```
POST /ngsi-ld/v1/entities/ HTTP/1.1
Host: localhost:9090
Link: <https://uri.etsi.org/ngsi-ld/primer/store-context.jsonld>;rel="http://www.w3.org/ns/json-ld#context";type="application/ld+json"
Content-Type: application/json

{
  "id": "urn:ngsi-ld:Shelf:123",
  "type": "Shelf",
  "maxCapacity": {
    "type": "Property",
    "value": 100
  }
}
```

```
}
}
```

**Figure 8.3: NGSi-LD Shelf entity creation using link header**

```
HTTP/1.1 201 Created
Date: Mon, 09 Sep 2019 13:20:23 GMT
location: /ngsi-ld/v1/entities/"urn:ngsi-ld:Shelf:123"
```

**Figure 8.4: NGSi-LD Shelf entity creation result**

## 8.2.2 Delete Entity

```
DELETE /ngsi-ld/v1/entities/urn:ngsi-ld:Store:002 HTTP/1.1
Host: localhost:9090
```

**Figure 8.5: NGSi-LD Store entity deletion**

Figure 8.5 shows the HTTP request for deleting the Store entity with the id `urn:ngsi-ld:Store:002` and Figure 8.6 the result in the case of successful deletion.

```
HTTP/1.1 204 No Content
Date: Wed, 04 Apr 2019 11:47:18 GMT
```

**Figure 8.6: NGSi-LD Store entity deletion result**

**NOTE:** In the following, the present document assumes that the entity with identifier `urn:ngsi-ld:Store:002` exists, so if it is deleted, it should be created again as shown in Figure 8.1.

## 8.3 Creating/Updating/Deleting Properties and Relationships

### 8.3.1 Appending an Entity Property/Relationship

Entities that have been created are not static but change over time. This includes adding new properties and relationships as opposed to updating the value or object of existing properties and relationships respectively. In the use case, a shelf may be put up in a store and this fact can be modelled as a contains relationship. Figure 8.7 shows the append operation that adds the contains relation to the store entity with id `urn:ngsi-ld:Store:001` that points to the shelf entity `urn:ngsi-ld:Shelf:123`. The successful result is shown in Figure 8.8.

```
POST /ngsi-ld/v1/entities/urn:ngsi-ld:Store:001/attrs HTTP/1.1
Host: localhost:9090
Content-Type: application/json
Link: <https://uri.etsi.org/ngsi-ld/primer/store-context.jsonld>;rel="http://www.w3.org/ns/json-ld#context";type="application/ld+json"

{
  "contains":{
    "type":"Relationship",
    "object":"urn:ngsi-ld:Shelf:123",
    "observedAt":"2019-09-09T12:09:07Z"
  }
}
```

**Figure 8.7: NGSi-LD append Store entity relationship**

```
HTTP/1.1 204 No Content
Date: Mon, 09 Sep 2019 14:08:15 GMT
```

**Figure 8.8: NGSi-LD append Store entity relationship result**

Relationships in NGS-LD are unidirectional, i.e. if the inverse relationship is required, then the API specification requires that it has to be explicitly created. In Figure 8.9, the `isContainedIn` relationship is appended to the shelf with `id urn:ngsi-ld:Shelf:123` that points to the store entity with `id urn:ngsi-ld:Store:001`, thus it represents the inverse of the `contains` relationship added to the store entity as shown in Figure 8.7. As request in Figure 8.9 is the same operation, the result generally looks the same as in Figure 8.8.

```
POST /ngsi-ld/v1/entities/urn:ngsi-ld:Shelf:123/attrs HTTP/1.1
Host: localhost:9090
Content-Type: application/json
Link: <https://uri.etsi.org/ngsi-ld/primer/store-context.jsonld>;rel="http://www.w3.org/ns/json-ld#context";type="application/ld+json"

{
  "isContainedIn":{
    "type":"Relationship",
    "object":"urn:ngsi-ld:Store:001",
    "observedAt":"2019-09-09T14:09:07Z"
  }
}
```

**Figure 8.9: NGS-LD append Shelf entity relationship using link header**

### 8.3.2 Updating an Entity Property/Relationship

If an existing property or relationship is to be updated, there is a corresponding NGS-LD operation. An example is shown in Figure 8.10. In the example the name of the store has changed from "Checker Market" to "Mega Market", thus the property `storeName` of the store entity with `id urn:ngsi-ld:Store:001` is updated accordingly. Figure 8.11 shows the successful result.

```
PATCH /ngsi-ld/v1/entities/urn:ngsi-ld:Store:001/attrs HTTP/1.1
Host: localhost:9090
Content-Type: application/json
Link: <https://uri.etsi.org/ngsi-ld/primer/store-context.jsonld>;rel="http://www.w3.org/ns/json-ld#context";type="application/ld+json"

{
  "storeName":{
    "type":"Property",
    "value":"Mega Market"
  }
}
```

**Figure 8.10: NGS-LD property update using link header**

```
HTTP/1.1 204 No Content
Date: Fri, 02 Aug 2019 15:20:13 GMT
```

**Figure 8.11: NGS-LD property update result**

### 8.3.3 Partial Update of an Entity Property/Relationship

If only parts of a property or relationship are to be updated, there is also the possibility of a partial update, where only a fragment of the attribute with the elements to be updated is provided. Figure 8.12 shows the update of the name on `storeName` resource representing the attribute, where only the changed value and not the complete property information is provided, unlike the example in Figure 8.10 where the complete attribute is updated. Figure 8.13 shows the successful result.

```
PATCH /ngsi-ld/v1/entities/urn:ngsi-ld:Store:001/attrs/storeName HTTP/1.1
Host: localhost:9090
Content-Type: application/json
Link: <https://uri.etsi.org/ngsi-ld/primer/store-context.jsonld>;rel="http://www.w3.org/ns/json-ld#context";type="application/ld+json"

{
  "value": "Mega Store"
}
```

**Figure 8.12: NGS-LD partial update of storeName entity property using link header**

```
HTTP/1.1 204 No Content
Date: Mon, 09 Sep 2019 12:45:41 GMT
```

**Figure 8.13: NGSi-LD partial update of storeName entity update result**

In Figure 8.14, an example is shown that not only the value of a property or the object of a relationship can be updated with a partial update, but also meta information, e.g. represented as a property of a property. In the example, the observedAt temporal property of the contains relationship that was appended in Figure 8.7 is updated, e.g. to make it consistent with the observedAt property of the inverse relationship added in Figure 8.9. Figure 8.15 shows the successful result.

```
PATCH /ngsi-ld/v1/entities/urn:ngsi-ld:Store:001/attrs/contains HTTP/1.1
Host: localhost:9090
Content-Type: application/json
Link: <https://uri.etsi.org/ngsi-ld/primer/store-context.jsonld>; rel="http://www.w3.org/ns/json-ld#context"; type="application/ld+json"

{
  "observedAt": "2019-09-09T14:09:07Z"
}
```

**Figure 8.14: NGSi-LD partial update of property of contains relationship using link header**

```
HTTP/1.1 204 No Content
Date: Mon, 09 Sep 2019 14:57:27 GMT
```

**Figure 8.15: NGSi-LD partial update of property of contains relationship result**

### 8.3.4 Deleting an Entity Property/Relationship

If a property or relationship is no longer valid, it can be deleted. In the example in Figure 8.16, the shelf has been removed from the store and thus the contains relationship of the store entity with id urn:ngsi-ld:Store:001 is deleted. Figure 8.17 shows the successful result. In case there is an inverse relationship, it has to be deleted separately.

```
DELETE /ngsi-ld/v1/entities/urn:ngsi-ld:Store:001/attrs/contains HTTP/1.1
Host: localhost:9090
Link: <https://uri.etsi.org/ngsi-ld/primer/store-context.jsonld>; rel="http://www.w3.org/ns/json-ld#context"; type="application/ld+json"
```

**Figure 8.16: NGSi-LD delete relationship contains using link header**

```
HTTP/1.1 204 No Content
Date: Mon, 09 Sep 2019 15:10:35 GMT
```

**Figure 8.17: NGSi-LD delete relationship contains result**

## 9 NGSi-LD Information Consumption

### 9.1 Overview NGSi-LD Information Consumption

Clause 9 describes the NGSi-LD API operations for information consumption, i.e. the operations that can be used to synchronously request entity information. Clause 9.2 describes the operations for retrieving information about a known entity, i.e. based on the entity identifier. Clause 9.3 describes the operations for querying entity information based on entity type, i.e. discovering new entities, filtering according to property values or relationship objects. Clause 9.4 introduces geographical queries, i.e. discovering entities of a certain type whose (current) location is within the specified geographic area.

## 9.2 Retrieving NGSI-LD Information

### 9.2.1 Retrieving NGSI-LD Entity Using Simplified Representation

In clause 5.4 an example is given how the full representation of an entity can be retrieved. Such a representation can include additional information about properties and relationships, e.g. properties of properties, typically used for meta information like a timestamp or the source of the information. If such meta information is not required, NGSI-LD also offers a simplified representation that only provides the property with its value or the relationship with its object in a key value presentation. In order to retrieve an entity in this representation, `keyValues` needs to be specified as an option. Figure 9.1 shows an example of the request for retrieving the entity with the entity identifier `urn:ngsi-ld:Store:001` with `options=keyValues`, using link header for providing the `@context`.

```
GET /ngsi-ld/v1/entities/urn:ngsi-ld:Store:001?options=keyValues HTTP/1.1
Host: localhost:9090
Accept: application/json
Link: <https://uri.etsi.org/ngsi-ld/primer/store-context.jsonld>;rel="http://www.w3.org/ns/json-ld#context";type="application/ld+json"
```

**Figure 9.1: NGSI-LD retrieval of simplified representation of Store:001 entity**

```
HTTP/1.1 200 OK
Date: Wed, 03 Apr 2019 15:50:09 GMT
Link: <https://uri.etsi.org/ngsi-ld/primer/store-context.jsonld>;rel="http://www.w3.org/ns/json-ld#context";type="application/ld+json"
Content-Type: application/json

{
  "id": "urn:ngsi-ld:Store:001",
  "type": "Store",
  "address": {
    "addressLocality": "Duckburg",
    "addressRegion": "Metropolis",
    "postalCode": "42000",
    "streetAddress": "Main Street 65"
  },
  "location": {
    "type": "Point",
    "coordinates": [13.3986, 52.5547]
  },
  "storeName": "Mega Store"
}
```

**Figure 9.2: NGSI-LD retrieved result with simplified representation of Store:002 entity**

Figure 9.2 shows the simplified representation of NGSI-LD Store:001 entity with only the values for the address, location and storeName properties.

## 9.3 NGSI-LD Query Language

### 9.3.1 Querying NGSI-LD Entities by Type

This example queries all NGSI-LD entities of type `Store` (mapped to `https://uri.etsi.org/ngsi-ld/primer/Store` in the `@context`) including all available properties and relationships. The request is shown in Figure 9.3 and the result with two `Store` entities in Figure 9.4. A link header is used for providing the `@context`.

```
GET /ngsi-ld/v1/entities?type=Store HTTP/1.1
Host: localhost:9090
Link: <https://uri.etsi.org/ngsi-ld/primer/store-context.jsonld>;rel="http://www.w3.org/ns/json-ld#context";type="application/ld+json"
Accept: application/json
```

**Figure 9.3: NGSI-LD query for entities by type**

```

HTTP/1.1 200 OK
Date: Wed, 05 Apr 2019 16:20:11 GMT
Link: <https://uri.etsi.org/ngsi-ld/primer/store-context.jsonld>;rel="http://www.w3.org/ns/json-ld#context";type="application/ld+json"
Content-Type: application/json

[
  {
    "id": "urn:ngsi-ld:Store:001",
    "type": "Store",
    "location": {
      "type": "GeoProperty",
      "value": {
        "type": "Point",
        "coordinates": [
          57.4874121,
          -20.2845607
        ]
      }
    },
    "address": {
      "type": "Property",
      "value": {
        "addressRegion": "Metropolis",
        "postalCode": "42000",
        "streetAddress": "Main Street 65",
        "addressLocality": "Duckburg"
      }
    },
    "storeName": {
      "type": "Property",
      "value": "Mega Store"
    }
  },
  {
    "id": "urn:ngsi-ld:Store:002",
    "type": "Store",
    "location": {
      "type": "GeoProperty",
      "value": {
        "type": "Point",
        "coordinates": [
          57.5522,
          -20.3484
        ]
      }
    },
    "address": {
      "type": "Property",
      "value": {
        "addressRegion": "Metropolis",
        "postalCode": "42420",
        "streetAddress": "Tiger Street 4",
        "addressLocality": "Cat City"
      }
    },
    "storeName": {
      "type": "Property",
      "value": "6-Stars"
    }
  }
]

```

**Figure 9.4: NGSI-LD query for entities by type results**

### 9.3.2 Querying NGSI-LD Entities by Type, Filtering by Property Value

Instead of retrieving all entities of a certain type, the results can be filtered using the `q` parameter. The filter can be on a simple property value as shown in Figure 9.5, where all entities of type `Store` (mapped to `https://uri.etsi.org/ngsi-ld/primer/Store` in the `@context`) that have a property `storeName` (mapped to `https://uri.etsi.org/ngsi-ld/primer/storeName` in the `@context`) whose value is "6-Stars".

```
GET /ngsi-ld/v1/entities?type=Store&q=storeName=="6-Stars" HTTP/1.1
Host: localhost:9090
Content-Type: application/json
Accept: application/json
Link: <https://uri.etsi.org/ngsi-ld/primer/store-context.jsonld>;rel="http://www.w3.org/ns/json-ld#context";type="application/ld+json"
```

**Figure 9.5: NGSi-LD query for entities by type, filtered by storeName**

As only a single store with the given storeName was created, only this store with the entity identifier `urn:ngsi-ld:Store:002` is returned as shown in Figure 9.6.

```
HTTP/1.1 200 OK
Date: Mon, 24 Feb 2020 10:44:30 GMT
Link: <https://uri.etsi.org/ngsi-ld/primer/store-context.jsonld>;rel="http://www.w3.org/ns/json-ld#context";type="application/ld+json"
Content-Type: application/json

[
  {
    "id": "urn:ngsi-ld:Store:002",
    "type": "Store",
    "location": {
      "type": "GeoProperty",
      "value": {
        "type": "Point",
        "coordinates": [
          57.5522,
          -20.3484
        ]
      }
    },
    "address": {
      "type": "Property",
      "value": {
        "addressRegion": "Metropolis",
        "postalCode": "42420",
        "streetAddress": "Tiger Street 4",
        "addressLocality": "Cat City"
      }
    },
    "storeName": {
      "type": "Property",
      "value": "6-Stars"
    }
  }
]
```

**Figure 9.6: NGSi-LD query for entities by type, filtered by storeName result**

It is also possible to filter according to elements of complex values. In Figure 9.7 the filter is according to the `addressRegion` element of `address`. Since both example stores have an address with `addressRegion=="Metropolis"`, the result is the same as in Figure 9.4.

```
GET /ngsi-ld/v1/entities?type=Store&q=address[addressRegion]=="Metropolis" HTTP/1.1
Host: localhost:9090
Accept: application/json
Link: <https://uri.etsi.org/ngsi-ld/primer/store-context.jsonld>;rel="http://www.w3.org/ns/json-ld#context";type="application/ld+json"
```

**Figure 9.7: NGSi-LD query for entities by type, filtered by addressRegion element of address**

**NOTE:** It is generally recommended, and in some cases required, to URL-encode (also referred to as percent encode) any parameter values in HTTP requests [i.3]. For readability reasons, this has not been done here, but the URL-encoded version of the filter would be:

```
address%5BaddressRegion%5D%3D%3D%22Metropolis%22
```

The NGSi-LD query language used to specify filters not only supports the 'equal' operator used in Figure 9.7, but all operators specified in Table 9.1.

**Table 9.1: Operators supported by the NGSI-LD query language**

Operator	Name	Comment
==	Equal	Comparison of numbers, quoted strings, dateTime, date and time, boolean, list of values (comma separated), range (dots, e.g. 1..5), URIs
!=	Unequal	Comparison of numbers, quoted strings, dateTime, date and time, boolean, list of values (comma separated), range (dots, e.g. 1..5), URIs
>	Greater	Comparison of numbers, quoted strings, dateTime, date and time
>=	Greater equal	Comparison of numbers, quoted strings, dateTime, date and time
<	Less	Comparison of numbers, quoted strings, dateTime, date and time
<=	Less equal	Comparison of numbers, quoted strings, dateTime, date and time
~=	Pattern	Comparison to a regular expression as described in IEEE POSIX 1003.2™ [i.4]
!~=	Not pattern	Comparison to a regular expression as described in IEEE POSIX 1003.2™ [i.4]

```
GET /ngsi-ld/v1/entities?type=https://w3id.org/ngsi-ld/store/Store&q=address[addressRegion]=="Metropolis";address[addressLocality]=="Cat City" HTTP/1.1
Host: localhost:9090
Accept: application/json
Link: <https://uri.etsi.org/ngsi-ld/primer/store-context.jsonld>;rel="http://www.w3.org/ns/json-ld#context";type="application/ld+json"
```

**Figure 9.8: NGSI-LD query for entities by type with logical AND of two filter elements**

It is also possible to combine multiple filters with the logical operators AND (;) or OR (). Figure 9.8 shows an example where both `address[addressRegion]=="Metropolis"` and `(;) address[addressLocality]=="Cat City"` have to be true. This is only the case for the store with the entity identifier `urn:ngsi-ld:Store:002`, thus the result is the same as in Figure 9.6.

### 9.3.3 Querying NGSI-LD Entities by Type, Filtering by Relationship Object

Not only properties, but also relationships can be used for filtering. Figure 9.9 shows an example, where all shelves are requested that have an `isContainedIn` relationship to the store with URI `urn:ngsi-ld:Store:001`. The result is one shelf with the URI `urn:ngsi-ld:Shelf:123` as entity identifier as shown in Figure 9.10.

```
GET /ngsi-ld/v1/entities?type=Shelf&q=isContainedIn==urn:ngsi-ld:Store:001 HTTP/1.1
Host: localhost:9090
Accept: application/json
Link: <https://uri.etsi.org/ngsi-ld/primer/store-context.jsonld>;rel="http://www.w3.org/ns/json-ld#context";type="application/ld+json"
```

**Figure 9.9: NGSI-LD query for entities by type, filtered by relationship**

```
HTTP/1.1 200 OK
Mon, 24 Feb 2020 10:49:58 GMT
Link: <https://uri.etsi.org/ngsi-ld/primer/store-context.jsonld>;rel="http://www.w3.org/ns/json-ld#context";type="application/ld+json"
Content-Type: application/json

[
  {
    "id": "urn:ngsi-ld:Shelf:123",
    "type": "Shelf",
    "isContainedIn": {
      "type": "Relationship",
      "object": "urn:ngsi-ld:Store:001",
      "observedAt": "2019-09-09T14:09:07Z"
    },
    "maxCapacity": {
      "type": "Property",
      "value": 100
    }
  }
]
```

**Figure 9.10: NGSI-LD query for entities by type, filtered by relationship result**

### 9.3.4 Querying NGSI-LD Entities by Type, Filtering by Meta Information

It is also possible to filter by meta information, i.e. properties of properties, relationships of properties, properties of relationships and relationships of relationships. In Figure 9.11 an example is given, where only shelf entities with `isContainedIn` relationships are considered that themselves have a temporal property `observedAt` that is greater or equal to `2019-09-09T14:09:07Z`. As `observedAt` is equal to `2019-09-09T14:09:07Z` and there are no other shelf entities, the result is the same as in Figure 9.10. If only values for `observedAt` greater than `2019-09-09T14:09:07Z` are considered as in Figure 9.12, the result will be the empty list.

```
GET /ngsi-ld/v1/entities?type=Shelf&q=isContainedIn.observedAt>=2019-09-09T14:09:07Z HTTP/1.1
Host: localhost:9090
Accept: application/json
Link: <https://uri.etsi.org/ngsi-ld/primer/store-context.jsonld>;rel="http://www.w3.org/ns/json-ld#context";type="application/ld+json"
```

**Figure 9.11: NGSI-LD query for entities by type, filtered by meta information**

```
GET /ngsi-ld/v1/entities?type=Shelf&q=isContainedIn.observedAt>2019-09-09T14:09:07Z HTTP/1.1
Host: localhost:9090
Accept: application/json
Link: <https://uri.etsi.org/ngsi-ld/primer/store-context.jsonld>;rel="http://www.w3.org/ns/json-ld#context";type="application/ld+json"
```

**Figure 9.12: NGSI-LD query for entities by type, filtered by meta information leading to empty result**

## 9.4 Geographical NGSI-LD Queries

In addition to filtering by property values, geographic queries allow the filtering of entities based on their geographic location. NGSI-LD supports special properties of type `GeoProperty` that are required by the specification to be specified in GeoJson [i.5]. With the geographic query language, a geographic area is specified that is matched against the specified `GeoProperty` (if no `GeoProperty` is specified, the default is `location`). Figure 9.13 gives an example, where the geographic query specified a point location given as a GPS coordinate and a radius of 1 000 metres around this coordinate. The result in Figure 9.14 contains the Store entity with entity identifier `urn:ngsi-ld:Store:002`, whose location is within the specified geographic area.

```
GET /ngsi-ld/v1/entities?type=Store&geoproperty=location&georel=near;maxDistance==1000&geometry=Point&coordinates=[57.5522023,-20.34840123] HTTP/1.1
Host: localhost:9090
Accept: application/json
Link: <https://uri.etsi.org/ngsi-ld/primer/store-context.jsonld>;rel="http://www.w3.org/ns/json-ld#context";type="application/ld+json"
```

**Figure 9.13: NGSI-LD query for entities by type, filtered by geographical query**

```
HTTP/1.1 200 OK
Mon, 24 Feb 2020 10:53:41 GMT
Link: <https://uri.etsi.org/ngsi-ld/primer/store-context.jsonld>;rel="http://www.w3.org/ns/json-ld#context";type="application/ld+json"
Content-Type: application/json

[
  {
    "id": "urn:ngsi-ld:Store:002",
    "type": "Store",
    "location": {
      "type": "GeoProperty",
      "value": {
        "type": "Point",
        "coordinates": [57.5522,-20.3484]
      }
    },
    "address": {
      "type": "Property",
      "value": {
        "addressRegion": "Metropolis",
        "postalCode": "42420",
        "streetAddress": "Tiger Street 4",
        "addressLocality": "Cat City"
      }
    }
  }
]
```

```

    }
  },
  "storeName": {
    "type": "Property",
    "value": "6-Stars"
  }
}
]

```

Figure 9.14: NGSi-LD query for entities by type, filtered by geographic query result

## 10 NGSi-LD Information Subscriptions

### 10.1 Overview NGSi-LD Information Subscriptions

Clause 10 describes the NGSi-LD API operations for subscriptions and the resulting notifications, i.e. the operations that can be used to request entity information and asynchronously receive the information in form of notifications. Clause 10.2.1 describes change-based subscriptions and how updates and the creation of new entities trigger notifications. Clause 10.2.2 describes time-based subscriptions and notifications that are triggered whenever the specified time interval has passed, independent of any changes to the entity information.

### 10.2 Creating Subscriptions

#### 10.2.1 Change-based Subscriptions, Updates and Resulting Notifications

Change-based subscription are based on the change of an attribute, i.e. the value of a property or the object of a relationship. The attributes whose changes trigger the notification are specified as a list in the `watchedAttributes` element of a subscription. Only in case this attribute changes in an existing entity or a new entity with this attribute is created, a notification will be sent.

In Figure 10.1, a subscription for entities of type `Store` (mapped to `https://uri.etsi.org/ngsi-ld/primer/Store` in the `@context`) and `watchedAttributes` with the property `storeName` (mapped to `https://uri.etsi.org/ngsi-ld/primer/storeName` in the `@context`) is shown.

The notification to be sent on a change is further specified in the `notification` element. The attributes specified are those to be included for the entities to be returned, if available. If no attribute is available for an entity, it will not be returned. If no attributes are specified, all available attributes will be returned.

For the avoidance of any doubt, for the attributes to be returned, only those specified in `attributes` will be considered. These can include the triggering attribute from `watchedAttributes` or not – the triggering attribute is not always returned, only in case it is specified. In the example in Figure 10.1, the properties `storeName` and `address` (but not `location`), will be returned, if available.

```

POST /ngsi-ld/v1/subscriptions HTTP/1.1
Host: localhost:9090
Content-Type: application/json
Link: <https://uri.etsi.org/ngsi-ld/primer/store-context.jsonld>; rel="http://www.w3.org/ns/json-ld#context"; type="application/ld+json"

{
  "id": "urn:ngsi-ld:Subscription:storeSubscription",
  "type": "Subscription",
  "entities": [
    {
      "type": "Store"
    }
  ],
  "watchedAttributes": ["storeName"],
  "notification": {
    "attributes": ["storeName", "address"],
    "format": "normalized",
    "endpoint": {

```

```

    "uri": "http://localhost:8000/notify",
    "accept": "application/json"
  }
}

```

**Figure 10.1: NGSi-LD subscription to entities of type Store, based on changes**

For the format, either `normalized` (which is the default) or `keyValue` can be specified. Just as in the case of queries, `keyValue` will only return pairs of attribute names and values without any additional information, otherwise the complete information will be returned.

In the endpoint element, the `uri` specifies HTTP URL to which the notification is to be sent. This requires that the URL is reachable with an HTTP request, i.e. cannot be behind a firewall. The `accept` specifies the JSON-LD format, i.e. valid mime-types are `application/json` (i.e. the `@context` will be provided in a Link header) and `application/ld+json` (i.e. the `@context` will be provided as an element in the JSON body).

```

PATCH /ngsi-ld/v1/entities/urn:ngsi-ld:Store:001/attrs/storeName HTTP/1.1
Host: localhost:9090
Content-Type: application/json
Link: <https://uri.etsi.org/ngsi-ld/primer/store-context.jsonld>; rel="http://www.w3.org/ns/json-ld#context"; type="application/ld+json"

{
  "value": "Mega Store"
}

```

**Figure 10.2: NGSi-LD partial update of storeName property in urn:ngsi-ld:Store:001 entity**

When updating the name of the Store entity with entity identifier `urn:ngsi-ld:Store:001` from "Checker Market" to "Mega Store" as shown in Figure 10.2, the subscription `urn:ngsi-ld:Subscription:storeSubscription` matches as the type is `Store` and the updated property `storeName` matches an element in the `watchedAttributes` list. Thus, the notification in Figure 10.3 is triggered.

```

POST /notify HTTP/1.1
Host: localhost:8000
Link: <https://uri.etsi.org/ngsi-ld/primer/store-context.jsonld>; rel="http://www.w3.org/ns/json-ld#context"; type="application/ld+json"
Content-Type: application/json
{
  "id": "ngsildbroker:notification:-4999875641436467960",
  "type": "Notification",
  "data": [
    {
      "id": "urn:ngsi-ld:Store:001",
      "type": "Store",
      "address": {
        "type": "Property",
        "createdAt": "2020-02-05T09:04:40.987000Z",
        "value": {
          "address:Locality": "Duck Village",
          "addressRegion": "Metropolis",
          "postalCode": "42000",
          "streetAddress": "Main Street 65"
        },
        "modifiedAt": "2020-02-05T09:04:40.987000Z"
      },
      "storeName": {
        "type": "Property",
        "createdAt": "2020-02-05T09:04:40.987000Z",
        "value": "Mega Store",
        "modifiedAt": "2020-02-05T10:36:22.691000Z"
      }
    }
  ],
  "notifiedAt": "2020-02-05T10:36:23.709000Z",
  "subscriptionId": "urn:ngsi-ld:Subscription:storeSubscription"
}

```

**Figure 10.3: NGSi-LD notification to subscription triggered by update of storeName property**

The notification contains the `subscriptionId`, so it is always clear which subscription triggered the notification and a `notifiedAt` timestamp.

The entity is contained in the `data` element of the notification and, as specified, includes that properties `address` and `storeName`, including `modifiedAt` timestamp of the `storeName`, i.e. when the property was updated.

```
POST /ngsi-ld/v1/entities/ HTTP/1.1
Host: localhost:9090
Content-Type: application/json
Link: <https://uri.etsi.org/ngsi-ld/primer/store-context.jsonld>; rel="http://www.w3.org/ns/json-ld#context"; type="application/ld+json"

{
  "id": "urn:ngsi-ld:Store:003",
  "type": "Store",
  "address": {
    "type": "Property",
    "value": {
      "streetAddress": "Lion Street 25",
      "addressRegion": "Metropolis",
      "addressLocality": "Cat City",
      "postalCode": "42420"
    }
  },
  "location": {
    "type": "GeoProperty",
    "value": {
      "type": "Point",
      "coordinates": [
        57.5722,
        -20.3584
      ]
    }
  },
  "storeName": {
    "type": "Property",
    "value": "Giga Market"
  }
}
```

**Figure 10.4: NGSi-LD creation of new Store entity with storeName Giga Market**

A notification for the subscription can also be triggered by the creation of a new entity as shown in Figure 10.4. The creation results in the notification in Figure 10.5, containing the entity with the entity identifier `urn:ngsi-ld:Store:003` and properties `address` and `storeName` as specified in the subscription.

```
POST /notify HTTP/1.1
Host: localhost:8000
Link: <https://uri.etsi.org/ngsi-ld/primer/store-context.jsonld>; rel="http://www.w3.org/ns/json-ld#context"; type="application/ld+json"
Content-Type: application/json
{
  "id": "ngsildbroker:notification:-8311420531544870425",
  "type": "Notification",
  "data": [
    {
      "id": "urn:ngsi-ld:Store:003",
      "type": "Store",
      "address": {
        "type": "Property",
        "createdAt": "2020-02-05T15:54:56.813000Z",
        "value": {
          "addressRegion": "Metropolis",
          "postalCode": "42420",
          "streetAddress": "Lion Street 25",
          "addressLocality": "Cat City"
        }
      },
      "modifiedAt": "2020-02-05T15:54:56.813000Z"
    },
    "storeName": {
      "type": "Property",
      "createdAt": "2020-02-05T15:54:56.813000Z",
      "value": "Giga Market",
      "modifiedAt": "2020-02-05T15:54:56.813000Z"
    }
  ]
}
```

```

    },
    "notifiedAt": "2020-02-05T15:54:57.024000Z",
    "subscriptionId": "urn:ngsi-ld:Subscription:storeSubscription"
  }
}

```

**Figure 10.5: NGSi-LD notification to subscription triggered by creation of new Store with storeName property**

## 10.2.2 Time-based Subscriptions and Resulting Notifications

Time-based subscriptions are sent after each notification interval, i.e. independent of whether there have been any changes in the entities that fit the subscription or not.

In Figure 10.6, a subscription for entities of type Store (mapped to `https://uri.etsi.org/ngsi-ld/primer/Store` in the `@context`) is shown. Notifications are to be sent every `timeInterval`, in this case every 60 seconds. The notification element is the same as in Figure 10.1, i.e. store entities with the attributes `storeName` and `address`, in normalized format are to be sent as HTTP POST requests to the notification endpoint that has the uri `http://localhost:8000/notify`.

```

POST /ngsi-ld/v1/subscriptions HTTP/1.1
Host: localhost:9090
Content-Type: application/json
Link: <https://uri.etsi.org/ngsi-ld/primer/store-context.jsonld>; rel="http://www.w3.org/ns/json-ld#context"; type="application/ld+json"

{
  "id": "urn:ngsi-ld:Subscription:intervalStoreSubscription",
  "type": "Subscription",
  "entities": [
    {
      "type": "Store"
    }
  ],
  "timeInterval": 60,
  "notification": {
    "attributes": ["storeName", "address"],
    "format": "normalized",
    "endpoint": {
      "uri": "http://localhost:8000/notify",
      "accept": "application/json"
    }
  }
}

```

**Figure 10.6: NGSi-LD subscription to entities of type Store, based on time**

In Figure 10.7, a notification sent as the result of the time-based subscription in Figure 10.6 is shown.

```

POST /notify HTTP/1.1
Host: localhost:8000
Link: <https://uri.etsi.org/ngsi-ld/primer/store-context.jsonld>; rel="http://www.w3.org/ns/json-ld#context"; type="application/ld+json"
Content-Type: application/json

{
  "id" : "ngsildbroker:notification:-6431227878819736798",
  "type" : "Notification",
  "data" : [ {
    "id" : "urn:ngsi-ld:Store:002",
    "type" : "Store",
    "address" : {
      "type" : "Property",
      "value" : {
        "addressRegion" : "Metropolis",
        "postalCode" : "42420",
        "streetAddress" : "Tiger Street 4",
        "addressLocality" : "Cat City"
      }
    }
  },
  "storeName" : {
    "type" : "Property",
    "value" : "6-Stars"
  }
}

```

```

    }, {
      "id" : "urn:ngsi-ld:Store:001",
      "type" : "Store",
      "address" : {
        "type" : "Property",
        "value" : {
          "address:Locality" : "Duck Village",
          "addressRegion" : "Metropolis",
          "postalCode" : "42000",
          "streetAddress" : "Main Street 65"
        }
      },
      "storeName" : {
        "type" : "Property",
        "value" : "Mega Market"
      }
    }, {
      "id" : "urn:ngsi-ld:Store:003",
      "type" : "Store",
      "address" : {
        "type" : "Property",
        "value" : {
          "addressRegion" : "Metropolis",
          "postalCode" : "42420",
          "streetAddress" : "Lion Street 25",
          "addressLocality" : "Cat City"
        }
      },
      "storeName" : {
        "type" : "Property",
        "value" : "Giga Market"
      }
    }
  ] ],
  "notifiedAt" : "2020-02-09T18:08:02.669000Z",
  "subscriptionId" : "urn:ngsi-ld:Subscription:intervalStoreSubscription"
}

```

**Figure 10.7: NGSi-LD notification to time-based subscription**

Subscriptions can also be retrieved as shown in Figure 10.8.

```

GET /ngsi-ld/v1/subscriptions/urn:ngsi-ld:Subscription:intervalStoreSubscription HTTP/1.1
Host: localhost:9090
Accept: application/json
Link: <https://uri.etsi.org/ngsi-ld/primer/store-context.jsonld>; rel="http://www.w3.org/ns/json-ld#context"; type="application/ld+json"

```

**Figure 10.8: NGSi-LD retrieval of a subscription**

In addition to the elements that were provided on creation, some status information is provided. In Figure 10.9, it can be seen when the last notification was sent, when the last successful notification was sent and how many notifications have been sent since the creation of the subscription. In case the sending of a notification failed, the time of the last failed attempt would also be shown.

```

{
  "id": "urn:ngsi-ld:Subscription:intervalStoreSubscription",
  "type": "Subscription",
  "entities": {
    "type": "Store"
  },
  "notification": {
    "attributes": [
      "storeName",
      "address"
    ],
    "endpoint": {
      "accept": "application/json",
      "uri": "http://localhost:8000/notify"
    },
    "format": "normalized",
    "lastNotification": "2020-02-09T18:13:02.676000Z",
    "lastSuccess": "2020-02-09T18:13:02.676000Z",
    "timesSent": 16
  },
  "timeInterval": 60
}

```

**Figure 10.9: Retrieved Subscription**

---

## Annex A: Change History

<b>Date</b>	<b>Version</b>	<b>Information about changes</b>
March 2018	0.0.1	Created ToC
September 2018	0.1.1	Provided initial content to clauses 4 to 8
January 2020	0.2.1	Short initial Primer ToC and agreed contributions
February 2020	0.3.1	Complete draft content for initial Primer version
March 2020	1.0.0	ETSI Technical Officer review for publication pre-processing after Final Draft ISG CIM Approval by remote consensus (ended and ratified on Sunday 1 <sup>st</sup> March 2020)
March 2020	1.1.1	editHelp Publication

---

## History

<b>Document history</b>		
V1.1.1	March 2020	Publication