ETSI GR CIM 008 V1.2.1 (2023-10)

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
RGR/CIM-008-NGSI-LD-PriV121

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 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Important notice

The present document can be downloaded from:
https://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.

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/CommiteeSupportStaff.aspx

If you find a security vulnerability in the present document, please report it through our
Coordinated Vulnerability Disclosure Program:
https://www.etsi.org/standards/coordinated-vulnerability-disclosure

Notice of disclaimer & limitation of liability

The information provided in the present deliverable is directed solely to professionals who have the appropriate degree of experience to understand and interpret its content in accordance with generally accepted engineering or other professional standard and applicable regulations.
No recommendation as to products and services or vendors is made or should be implied.
No representation or warranty is made that this deliverable is technically accurate or sufficient or conforms to any law and/or governmental rule and/or regulation and further, no representation or warranty is made of merchantability or fitness for any particular purpose or against infringement of intellectual property rights.
In no event shall ETSI be held liable for loss of profits or any other incidental or consequential damages.

Any software contained in this deliverable is provided "AS IS" with no warranties, express or implied, including but not limited to, the warranties of merchantability, fitness for a particular purpose and non-infringement of intellectual property rights and ETSI shall not be held liable in any event for any damages whatsoever (including, without limitation, damages for loss of profits, business interruption, loss of information, or any other pecuniary loss) arising out of or related to the use of or inability to use the software.

Copyright Notification

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are 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 Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, 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.

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.

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, replaced 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 data model example, where the data model conforms to the NGSI-LD information model. 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".

[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".

[i.6] ISG CIM Forge: "Postman Scripts for NGSI-LD Primer".

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 1: 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.

NOTE 2: The terms defined in this clause are capitalized throughout the present document.

NGSI-LD Attribute: reference to both an NGSI-LD Property and to an NGSI-LD Relationship

NGSI-LD Attribute Instance (in case of temporal representation of NGSI-LD Entities): reference to an NGSI-LD Attribute, at a specific moment in time of its temporal evolution, usually identified by its instanceId

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 Context Registry: software functional element where Context Sources register the information that they can provide

NOTE: It is used by Distribution Brokers and Federation Brokers to find the appropriate Context Sources which can provide the information required for serving an NGSI-LD request.

NGSI-LD Context Source: source of context information which implements the NGSI-LD consumption and subscription (and possibly provision) interfaces defined by the present document

NOTE: It is usually registered with an NGSI-LD Registry so that it can announce what kind of information it can provide, when requested, to Context Consumers and Brokers.

NGSI-LD Context Source Registrations: description of the information that can be provided by a Context Source, which is used when registering the Context Source with the Context Registry

NGSI-LD Core API: core part of the NGSI-LD API that has to be implemented by all Brokers, including operations for providing or managing Entities and Attributes, operations for consuming Entities and checking which Entity Types and Attributes Entities are available in the system and operations for subscribing to Entities, receiving notifications and managing subscriptions

NGSI-LD Distribution Broker: NGSI-LD Context Broker that uses both local context information and registration information from an NGSI-LD Context Registry, to access matching context information from a set of distributed Context Sources

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)**.

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".

EXAMPLE 3: Alice's motorhome has a unique URI as id, but can be assigned multiple NGSI-LD Entity Types, e.g. "Vehicle" and "Home".

NGSI-LD Federation Broker: Distribution Broker that federates information from multiple underlying NGSI-LD Context Brokers and across domains

NGSI-LD GeoProperty: subclass of NGSI-LD Property which is a 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, that uses the special *hasValue* Property to define its target value and holds a geographic location in GeoJSON Format

NGSI-LD Language Map: JSON-LD language map in the form of key-value pairs holding the string representation of a main characteristic in a series of natural languages

EXAMPLE: "Bob's vehicle is currently parked on a street which is known as 'Grand Place' in French and 'Grote Markt' in Dutch" can be represented by an NGSI-LD LanguageProperty whose Name is "street" which holds an NGSI-LD Language Map of two key-value pairs containing both the French ("fr") and Dutch ("nl") exonyms of the street name.

NGSI-LD LanguageProperty: subclass of NGSI-LD Property which is a description instance which associates a set of strings in different natural languages as a defined main characteristic, i.e. an **NGSI-LD Map**, to an NGSI-LD Entity, an NGSI-LD Relationship or another NGSI-LD Property and that uses the special *hasLanguageMap* (a subproperty of *hasValue*) Property to define its target value

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 Null: *"urn:ngsi-ld:null"* or *{"@none": "urn:ngsi-ld:null"}* used as an encoding for *null* values

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 Query: collection of criteria used to select a sub-set of NGSI-LD Entities, matching the criteria

NGSI-LD Registry API: part of the NGSI-LD API that is implemented by the Context Registry, including operations for registering Context Sources and managing Context Source Registrations (CSRs), operations for retrieving and discovering CSRs, and operations for subscribing to CSRs and receiving notifications

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 Scope: enables putting Entities into a hierarchical structure and scoping queries and subscriptions according to it

NGSI-LD Temporal API: part of the NGSI-LD API pertaining to the Temporal Evolution of Entities, including operations for providing and managing the Temporal Evolution of Entities and Attributes, and operations for consuming the Temporal Evolution of Entities

NGSI-LD Temporal Evolution of Entities: sequence of values attributed to them over time, i.e. their history or future predictions

NGSI-LD Tenant: user or a group of users that utilize a single instance of a system implementing the NGSI-LD API (NGSI-LD Context Source or NGSI-LD Broker) in isolation from other users or groups of users of the same instance. Any information related to one tenant (e.g. Entities, Subscriptions, Context Source Registrations) are only visible to users of the same tenant, but not to users of a different tenant

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).

NGSI-LD VocabProperty: subclass of NGSI-LD Property which is a description instance which associates a string value which can be coerced to a URI as a defined main characteristic, i.e. an NGSI-LD Vocabulary, to an NGSI-LD Entity, an NGSI-LD Relationship or another NGSI-LD Property and that uses the special *hasVocab* (a subproperty of *hasValue*) Property to define its target value

NGSI-LD Vocabulary: string representation of a main characteristic which is explicitly defined to undergo JSON-LD type coercion to a URI

EXAMPLE: "Bob's car is a non-commercial vehicle" can be represented by an NGSI-LD VocabProperty whose Name is "category" which holds an NGSI-LD Vocabulary with the string value *"non-commercial"*. If the associated JSON-LD context defines the term *"non-commercial"* as *"http://example.com/ non-commercial"*, then the returned value will be the expanded using type coercion into the IRI the *http://example.com/ non-commercial*.

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 and Relationships Entities can have, is determined by the Entity Type, which in turn can be defined as part of a data model.

The NGSI-LD API and the Context Brokers implementing it are only based on the abstract NGSI-LD information model, which defines the Entity concept, and that Entities can have Properties and Relationships. They are agnostic to the data model, i.e. what Entity Types exist and what concrete Properties and Relationship the Entity instances can have. Thus, Context Brokers cannot enforce the conformance to a specific data model, making this the responsibility of the users and their applications.

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 Entity instances of Entity Type (grocery) store "6-Stars" and "Checker Market" with its location depicted on a map, a product "Wine" and a customer "Paul".

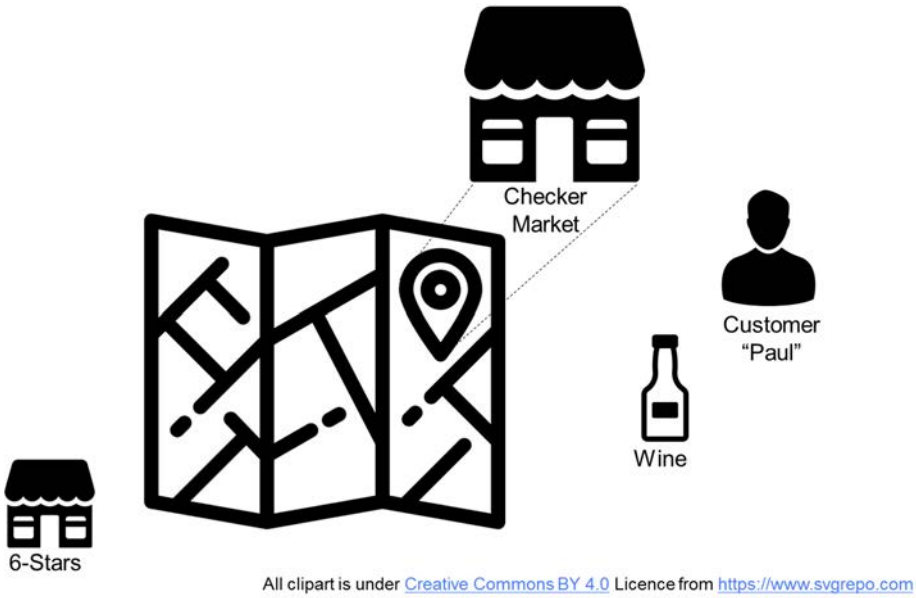

Figure 4-1: Grocery store use case example

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 Entity 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).

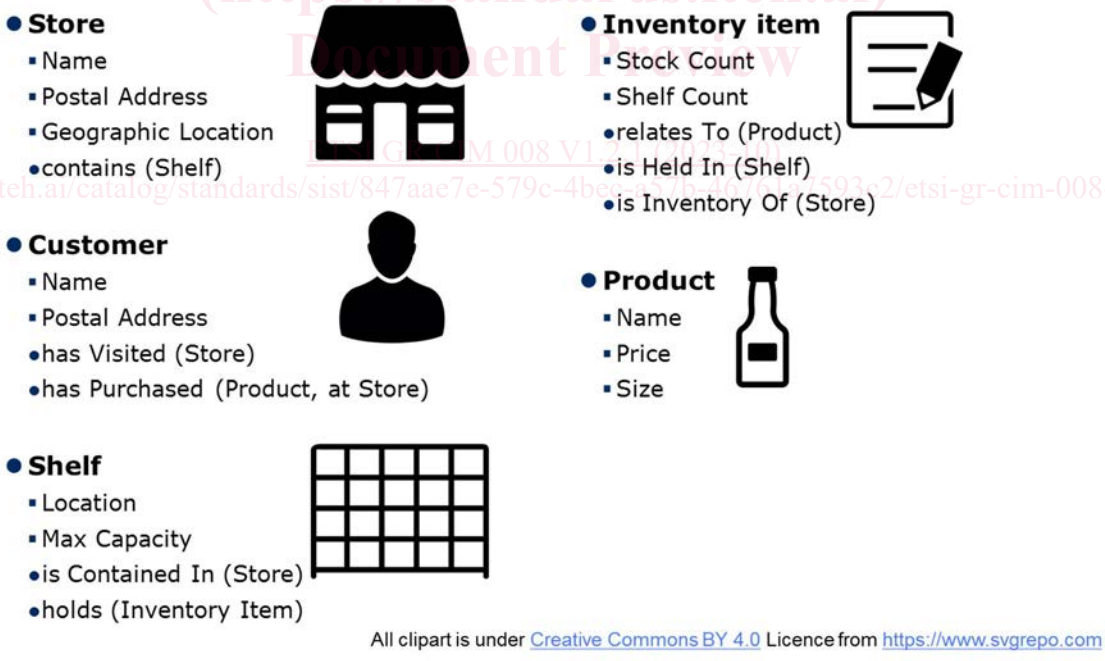

Figure 4-2: Entity Types, Relationships and Properties of use case

5 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. a Context Broker.

To make it easier to try out NGSI-LD, all examples in the present document are available online as Postman scripts [i.6]. Using one of the available (open-source) implementations of NGSI-LD Brokers, the interested reader can play around with the examples and get a hands-on experience of NGSI-LD.

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 Central Context Broker that stores all information. The resulting architecture is depicted in Figure 5.2-1 and requires a certain subset of NGSI-LD operations that are introduced in clauses 5, 6, 7, 8, 9 and 10.

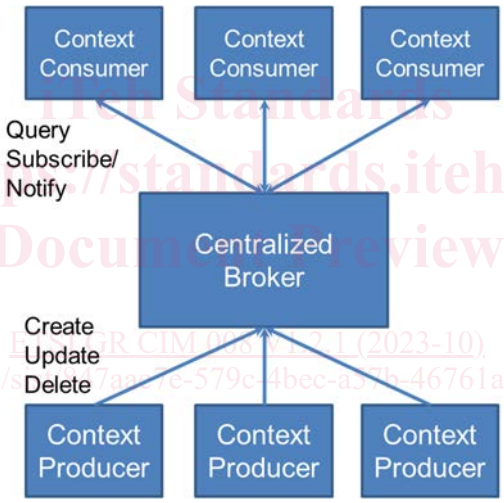

Figure 5.2-1: Basic architectural assumptions

The roles in this setup are Context Producers, Context Consumers and a Central Broker. The assumption is that the Central 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 Central 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 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`) as shown in Figure 5.3-1.

```
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-v1.7.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 5.3-1: Entity creation

If the creation was successful, the response in Figure 5.3-2 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 5.3-2: 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 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 5.4-1 and the successful result in Figure 5.4-2. Note that since no @context was provided in the request, only the core context is returned, whereas all user-defined aspects are returned as URIs [i.3]. The URIs prefixed with "`ngsi-ld`" appear instead of the full URIs, because the NGSI-LD Core context (`https://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context-v1.7.jsonld`), which is considered the default JSON-LD @context, if no other @context is specified, defines the `ngsi-ld` namespace for `https://uri.etsi.org/ngsi-ld`, i.e. on doing the JSON-LD compaction of the results by the Broker, this namespace is used. 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 5.4-1: Entity retrieval

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

{
  "id": "urn:ngsi-ld:Store:001",
  "type": "ngsi-ld:primer/Store",
  "location": {
    "type": "GeoProperty",
    "value": {
      "type": "Point",
      "coordinates": [ 57.4874121, -20.2845607 ]
    }
  },
  "ngsi-ld:primer/address": {
    "type": "Property",
    "value": {
      "ngsi-ld:primer/addressLocality": "Duck Village",
      "ngsi-ld:primer/addressRegion": "Metropolis",
      "ngsi-ld:primer/postalCode": "42000",
      "ngsi-ld:primer/streetAddress": "Main Street 65"
    }
  },
  "ngsi-ld:primer/storeName": {
    "type": "Property",
    "value": "Checker Market"
  },
  "@context": [ "https://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context-v1.7.jsonld" ]
}
```

Figure 5.4-2: 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, replaced 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. Clause 11 provides examples for Batch Operations, Clause 12 shows how to find out what types of entities and what attributes are actually available. Clause 13 gives examples of how the Temporal API of NGSI-LD can be used.

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 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 Entity Types, Relationships and Properties are modelled as subclasses of the core concepts Entity, Relationship and Property respectively. For our example use case, the Entity Types, 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 ETSI GS CIM 009 [i.1].