

# OMA NGSI Association concept

From FI-WARE Forge Wiki

Jump to: [navigation](#), [search](#)

**This is draft version 1.0. Please note that this is a beta functionality not extensively tested yet. Thus, the specification may change in the future.**

## Contents

- [1 Associations in FI-WARE NGSI 9/10](#)
  - [1.1 Abstract description of associations between entities and attributes](#)
  - [1.2 Representation of associations in FI-WARE NGSI](#)
  - [1.3 Registering Associations using NGSI-9](#)
  - [1.4 Retrieving Associations using NGSI-9](#)
- [2 Usage Examples](#)
  - [2.1 Scenario Overview](#)
  - [2.2 Registration](#)
  - [2.3 Discovery of Sources](#)
  - [2.4 Discovery of Targets](#)

## Associations in FI-WARE NGSI 9/10

The purpose of the association concept in FI-WARE NGSI is to enable a transition from device-level information to Thing-level information and vice versa. Associations describe relationships between these two levels of abstraction.

In what follows we first describe associations as an abstract concept. Subsequently, the representation of associations in the FI-WARE NGSI information model is defined, then the registration and retrieval of associations with NGSI operations is described, and finally a set of examples is provided.

### Abstract description of associations between entities and attributes

The information model of the OMA NGSI context interfaces, and therefore also of the FI-WARE NGSI binding, is based on the notion of entities and their attributes. Entities represent arbitrary physical objects, and attributes represent properties of these objects. To simplify the discussion, we will use the syntax `<entityId>.<attributeName>` to refer to the attribute named `<attributeName>` of the entity named `<entityId>`.

An *attribute association* is an ordered pair of Entity/Attribute combinations, written in the form `<entityId_1>.<attributeName_1> --> <entityId2>.<attributeName_2>`. The semantics of such an association is that any value of `<entityId_1>.<attributeName_1>` can be interpreted as a value of `<entityId2>.<attributeName_2>`. We emphasize that this is a unidirectional relation, i.e., values of

<entityId2>.<attributeName\_2> are not necessarily to be interpreted as values of <entityId\_1>.<attributeName\_1>.

An *entity association* is an ordered pair of entities, written in the form <entityId\_1> --> <entityId\_2>. The entity association implicitly represents all attribute associations of the form <entityId\_1>.<attributeName> --> <entityId\_2>.<attributeName>, for any possible attribute <attributeName>.

For associations <entityId\_1>.<attributeName\_1> --> <entityId2>.<attributeName\_2>, <entityId\_1> is called the *source entity* and <attributeName\_1> is denoted *source attribute* of the association. Furthermore, <entityId\_2> and <attributeName\_2> is called *target entity* and *target attribute* of the association, respectively.

## Representation of associations in FI-WARE NGSI

The NGSI data structure used to represent associations is the ContextRegistration structure.

### ContextRegistration structure

Element name	Element type	Optional	Description
EntityIdList	EntityId[1..unbound]	Yes	List of identifiers for the Context Entities being registered
ContextRegistrationAttribute	ContextRegistrationAttribute[0..unbound]	Yes	List of ContextAttributes and AttributeDomains which are made available through this registration
RegistrationMetadata	ContextMetadata[1..unbound]	Yes	Metadata characterizing this registration
ProvidingApplication	xsd:anyURI	No	URI identifying the application that provides the values of the context attributes for the target Context Entities

Any association information is represented in the the RegistrationMetadata field. For this purpose, a special metadata type is defined below.

The EntityIdList and ContextRegistrationAttribute fields can be omitted, as all necessary information on the association will be contained in the Metadata. If these fields are not empty, then they shall be interpreted in the standard way, i.e., that the URI specified in the ProvidingApplication points to a NGSI-10 context provider that can provide information about the specified entities and attributes.

OMA NGSI 9/10 defines ContextMetadata as follows.

### ContextMetadata structure

Element name	Element type	Optional	Description
Name	xsd:string	No	Name of the metadata
Type	xsd:anyUri	Yes	Indicates the type of the value field
Value	xsd:any	No	The actual value of the metadata

Associations are represented by contextMetadata instances whose type is set to *association*, and whose value is an instance of the following data structure.

### Association structure

Element name	Element type	Optional	Description
SourceEntityId	EntityId	No	The Id of this association's source entity
TargetEntityId	EntityId	No	The Id of this association's target entity
AttributeAssociationList	AttributeAssociation[]	Yes	List of pairs of source attribute and target attribute. If this field is not present, then the Association instance represents an entity association.

### AttributeAssociation structure

Element name	Element type	Optional	Description
SourceAttribute	xsd:string	No	The name of this association's source attribute
TargetAttribute	xsd:string	No	The name of this association's target attribute

In case an Association instance contains no AttributeAssociationList field, it represents the entity association between the given source entity <sourceEntityId> and the given target entity <targetEntityId>. Otherwise the Association instance represents a set of attribute associations. For each pair of source attribute <sourceAttribute> and target attribute <targetAttribute> that appears as an attributeAssociation in the AttributeAssociationList, the attribute association <sourceEntityId>.<sourceAttribute> --> <targetEntityId>.<targetAttribute> is represented.

Note that a single instance of Association can represent multiple attribute associations, but they all refer to the same pair of source and target entity. For representing associations between multiple entity pairs one has to use multiple instances of the Association structure.

## Registering Associations using NGSI-9

Associations can be registered with NGSI-9 in the same way as context providers are registered. For registering a set of associations, a registerContextRequest message that contains the corresponding ContextRegistration structures has to be sent.

## Retrieving Associations using NGSI-9

Associations are retrieved using the standard operations `DiscoverContextAvailability` and `SubscribeContextAvailability` defined in NGSI-9. These operations are used to ask for sets of entity/attribute combinations. A special type of operation scope is defined for indicating the set of associations expected to be returned.

Scope Type	Scope Value
IncludeAssociations	One of the following keywords: SOURCES, TARGETS, ALL, or NONE

The four possible scope values have the following semantics:

- **NONE** means that no associations are expected to be returned. This is the default value when no scope of type `IncludeAssociations` is present.
- **TARGETS** is used when the requestor wishes to find out about the *target set* of a given set of entity/attribute combinations. The target set of an entity/attribute combination set is recursively defined as the smallest set such that (1) the original entity/attribute combinations are part of the target set and (2) if there is an association `<sourceEntity>.<sourceAttribute> --> <targetEntity>.<targetAttribute>` and `<sourceEntity>.<sourceAttribute>` is in the target set, then also `<targetEntity>.<targetAttribute>` is in the target set. The **TARGETS** keyword is used to request any association between entity/attribute combinations from the target set.
- **SOURCES** is used when the requestor wishes to find out where and how information about a given set of entity/attribute combinations can be retrieved. The *source set* of a set of entity/attribute combinations is recursively defined as the smallest set such that (1) the original entity/attribute combinations are part of the source set and (2) if there is an association `<sourceEntity>.<sourceAttribute> --> <targetEntity>.<targetAttribute>` and `<targetEntity>.<targetAttribute>` is in the source set, then also `<sourceEntity>.<sourceAttribute>` is in the source set. The **SOURCES** keyword is used to request any association between entity/attribute combinations from the source set and, additionally, any context provider of any entity/attribute combination from the source set.
- **ALL** means that the union of the responses corresponding to the **SOURCES** and **TARGETS** keywords is expected.

## Usage Examples

### Scenario Overview

In this set of examples we consider a simple scenario to demonstrate the usage of associations. The main actor is an NGSI 9 server; in FI-WARE this role could be played by the Configuration Management GE from the IoT Service Enablement chapter.

The sequence of events is as follows:

- An application registers a provider of information on attribute *measurement* of entity *sensor5* using the `registerContext` operation of NGSI 9.
- Another application registers an association between *sensor5.measurement --> house3.temperature* using also the `registerContext` operation of NGSI 9.

- A discovery for *house3.temperature* is issued with scope SOURCES. In FI-WARE, the issuer could be an instance of the IoT Broker GE, who has received a request for information on the attribute *temperature* of entity *house3* and seeks for sources of this information. The NGSI 9 server returns both the association *sensor5.measurement* --> *house3.temperature* and the context provider of *sensor5.measurement*.
- A discovery for *sensor5.measurement* is issued with scope TARGETS. In FI-WARE, the issuer could also here be an instance of the IoT Broker GE, who has received an attribute value of *sensor5.measurement* and seeks to find out how to interpret this value. In this case the NGSI 9 server returns the association *sensor5.measurement* --> *house3.temperature*.

## Registration

The first registration announces a context provider and thus associations play no role in it. The request message body would look as follows.

```
<?xml version="1.0" encoding="UTF-8"?>
<registerContextRequest>
  <contextRegistrationList>
    <contextRegistration>
      <entityIdList>
        <entityId type="TemperatureSensor" isPattern="false">
          <id>sensor5</id>
        </entityId>
      </entityIdList>
      <contextRegistrationAttributeList>
        <contextRegistrationAttribute>
          <name>measurement</name>
          <isDomain>>false</isDomain>
        </contextRegistrationAttribute>
      </contextRegistrationAttributeList>
      <providingApplication>http://myHomeGateway22.org</providingApplication>
    </contextRegistration>
  </contextRegistrationList>
  <duration>PT1M</duration>
  <registrationId></registrationId>
</registerContextRequest>
```

The second registration announces the association. As mentioned above, the association is represented in the metadata while the entity and attribute lists are absent.

```
<?xml version="1.0" encoding="UTF-8"?>
<registerContextRequest>
  <contextRegistrationList>
    <contextRegistration>
      <registrationMetaData>
        <contextMetadata>
          <name>association1</name>
          <type>Association</type>
          <value>
            <sourceEntityId type="TemperatureSensor" isPattern="false">
              <id>Sensor5</id>
            </sourceEntityId>
            <targetEntityId type="House" isPattern="false">
              <id>house3</id>
            </targetEntityId>
          <AttributeAssociationList>
            <AttributeAssociation>
```

```

        <sourceAttribute>measurement</sourceAttribute>
        <targetAttribute>indoorTemperature</targetAttribute>
    </AttributeAssociation>
</AttributeAssociationList>
    <value>
    </contextMetadata>
</registrationMeta-data>
    <providingApplication>http://www.fi-ware.eu/NGSI/association</providingApplication>
    <contextRegistration>
    </contextRegistrationList>
    <duration>PT1M</duration>
</registerContextRequest>

```

## Discovery of Sources

To discover the sources of *house3.temperature*, the following request is issued.

```

<?xml version="1.0" encoding="UTF-8"?>
<discoverContextAvailabilityRequest>
    <entityIdList>
        <entityId type="House" isPattern="false">
            <id>house3</id>
        </entityId>
    </entityIdList>
    <attributeList>
        <attribute>indoorTemperature</attribute>
    </attributeList>
    <restriction>
        <scope>
            <operationScope>
                <scopeType>IncludeAssociations</scopeType>
                <scopeValue>SOURCES</scopeValue>
            </operationScope>
        </scope>
    </restriction>
</discoverContextAvailabilityRequest>

```

The response from the NGSI 9 server looks as follows.

```

<?xml version="1.0" encoding="UTF-8"?>
<discoverContextAvailabilityResponse>
    <contextRegistrationResponseList>
        <contextRegistrationResponse>
            <contextRegistration>
                <entityIdList>
                    <entityId type="TemperatureSensor" isPattern="false">
                        <id>sensor5</id>
                    </entityId>
                </entityIdList>
                <contextRegistrationAttributeList>
                    <contextRegistrationAttribute>
                        <name>measurement</name>
                        <isDomain>>false</isDomain>
                    </contextRegistrationAttribute>
                </contextRegistrationAttributeList>
                <providingApplication>http://myHomeGateway22.org</providingApplication>
            </contextRegistration>
        </contextRegistrationResponse>
    </contextRegistrationResponseList>

```

```

<contextRegistration>
  <registrationMetaData>
    <contextMetadata>
      <name>association1</name>
      <type>Association</type>
      <value>
        <sourceEntityId type="TemperatureSensor" isPattern="false">
          <id>sensor5</id>
        </sourceEntityId>
        <targetEntityId type="House" isPattern="false">
          <id>house3</id>
        </targetEntityId>
        <AttributeAssociationList>
          <AttributeAssociation>
            <sourceAttribute>measurement</sourceAttribute>
            <targetAttribute>indoorTemperature</targetAttribute>
          </AttributeAssociation>
        </AttributeAssociationList>
      </value>
    </contextMetadata>
  </registrationMetaData>
  <providingApplication>http://www.fi-ware.eu/NGSI/association</providingApplication>
</contextRegistration>
</contextRegistrationResponse>
</contextRegistrationResponseList>
</discoverContextAvailabilityResponse>

```

## Discovery of Targets

The request for target discovery deviates from the source discovery by the scope value.

```

<?xml version="1.0" encoding="UTF-8"?>
<discoverContextAvailabilityRequest>
  <entityIdList>
    <entityId type="TemperatureSensor" isPattern="false">
      <id>sensor5</id>
    </entityId>
  </entityIdList>
  <attributeList>
    <attribute>measurement</attribute>
  </attributeList>
  <restriction>
    <scope>
      <operationScope>
        <scopeType>IncludeAssociations</scopeType>
        <scopeValue>TARGETS</scopeValue>
      </operationScope>
    </scope>
  </restriction>
</discoverContextAvailabilityRequest>

```

The response here looks as follows.

```

<?xml version="1.0" encoding="UTF-8"?>
<discoverContextAvailabilityResponse>
  <contextRegistrationResponseList>
    <contextRegistrationResponse>
      <contextRegistration>
        <registrationMetaData>

```

```

<contextMetadata>
  <name>association1</name>
  <type>Association</type>
  <value>
    <sourceEntityId type="TemperatureSensor" isPattern="false">
      <id>sensor5</id>
    </sourceEntityId>
    <targetEntityId type="House" isPattern="false">
      <id>house3</id>
    </targetEntityId>
    <AttributeAssociationList>
      <AttributeAssociation>
        <sourceAttribute>measurement</sourceAttribute>
        <targetAttribute>indoorTemperature</targetAttribute>
      </AttributeAssociation>
    </AttributeAssociationList>
  </value>
</contextMetadata>
</registrationMeta-data>
<providingApplication>http://www.fi-ware.eu/NGSI/association</providingApplication>
</contextRegistration>
</contextRegistrationResponse>
</contextRegistrationResponseList>
</discoverContextAvailabilityResponse>

```