

# **NGSI-9 Context Management RESTful binding**

**V1.0**

Contributors: NEC, Orange, NSN, SAP, TID

# ChangeLog

Date	Versión	Author	Comment
December 13 <sup>th</sup> , 2013	0.4	Tobias Jacobs (NEC)	<p>Adding changelog table</p> <p>Harmonized the introductory description with NGSI 10 binding</p> <p>Made support of interaction with /contextEntities/{EntityID}/attributes and /contextEntityTypes/{typeName}/attributes optional.</p> <p>Moved the contents of chapter 5 into the beginning of chapter 4.</p>
December 13 <sup>th</sup> , 2013	1.0	Tobias Jacobs (NEC)	Removed comments & finalized layout

# 1. Introduction

This document outlines a RESTful binding of the OMA NGSI-9 standard. The specifications are based on Version 1.0 of the Context Management Interface of the Next Generation Service Interface (NGSI-9).

## 1.1 Design Principles of the Binding

The mapping of NGSI-10 functionality to a resource tree follows a twofold approach. On the one hand, there is one resource per NGSI-10 operation which supports the respective functionality by providing a POST operation. On the other hand, a number of additional resources support convenience functionality. The latter resource structure more closely follows the REST approach and typically supports more operations (GET PUT, POST, and DELETE). The operation scope of the GET operation on these resources can further be limited by a URI parameter.

The convenience functions typically only support a subset of the functionality of the corresponding NGSI operations. Nevertheless, they enable simpler and more straightforward access.

Throughout the document it is assumed that all data structures, as well as the input and output messages of NGSI-10 are represented by xml types. The definition of these types can be found in the xml schema files provided with this document.

## 1.2 Remarks on the data model and operations of NGSI-9 and NGSI-10

This subsection clarifies a number of details about the data model of NGSI-9 and NGSI-10. These details are discussed at this point because they have significant impact on the RESTful binding.

### 1.2.1 Relation between Entities and Context Elements

Entities are the first-class citizens of the data model. Each entity is unambiguously determined by an `EntityId`, which contains an `Id` String and a `Type`. According to the nature of the convenience functions in the RESTful binding, the uniqueness of the context elements is guaranteed by the `Id` element.

An Entity may contain multiple *attribute domains*. An attribute domain is a logical grouping of context information attributes. The `ContextElement` data structure contains context information about one Entity. It is a container for one or more attribute values and respective metadata. If the attribute domain is specified within a `ContextElement`, then it may contain only values and metadata of attribute from that domain. Thus, a `ContextElement` does not necessarily contain *all* available information about an Entity. The set of convenience function resources includes a type of resource for representing an Entity. The URI of that sort of resource is determined by the ID of the represented entity. Note that two Entities having the identical name but differing type will be represented by the same resource.

### 1.2.2 Attribute Value Instances

In typical IoT environments there can be multiple providers delivering values of the same attribute. Therefore, NGSI-10 allows the existence of multiple attributes having the same name. The way to distinguish these attribute values - called *value instances* in this document - is to include an ID in the form of metadata. More specifically, the ID of an attribute value is determined by the unique value of a metadata with Name "ID".

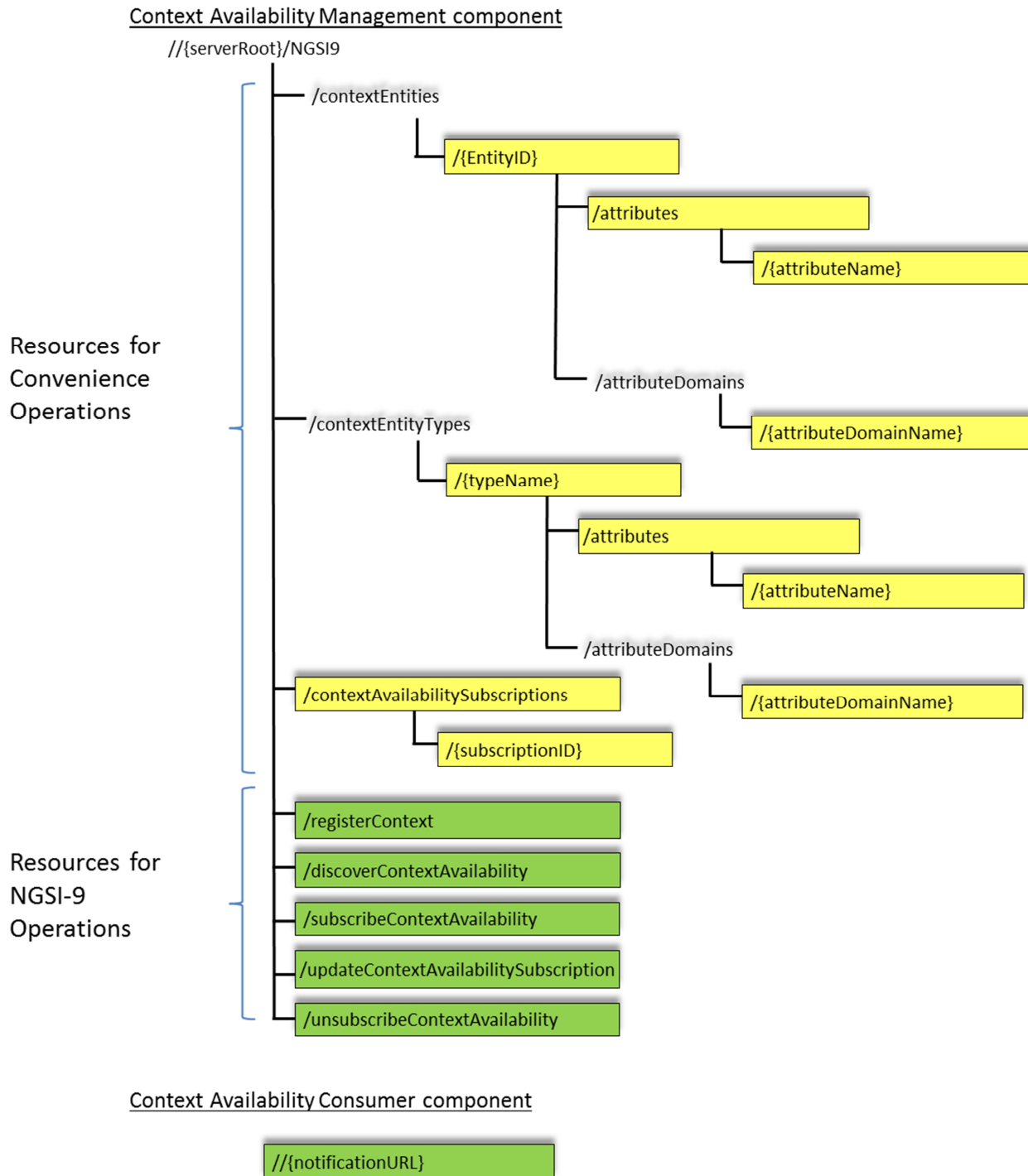
Each attribute value instance is represented by a convenience function resource in the RESTful binding.

In order to also enable simpler systems which do not distinguish between several attribute value instances, it is not mandatory for servers to enforce the usage of attribute value instances.

### 1.2.3 Subscriptions and existence of attribute values

In this document it is assumed that subscriptions can also be successful if they refer to entities and/or attributes the system does not have information about. Notifications are then sent as soon as the information becomes available and the condition for notification is satisfied.

## 2. Resources Summary



## 2.1 Resources on Context Management Component

### 2.1.1 Standard NGSI-9 Operation Resources

The five resources listed in the table below represent the five operations offered by systems that implement the NGSI-9 Context Management role. Each of these resources allows interaction via http POST. All attempts to interact by other verbs shall result in an HTTP error status 405; the server should then also include the 'Allow: POST' field in the response.

Resource	Base URI: http://{serverRoot}/NGSI9	HTTP verbs
		POST
Context registration resource	/registerContext	Generic context registration. The expected request body is an instance of <code>registerContextRequest</code> ; the response body is an instance of <code>registerContextResponse</code> .
Discovery resource	/discoverContextAvailability	Generic discovery of context information providers. The expected request body is an instance of <code>discoverContextAvailabilityRequest</code> ; the response body is an instance of <code>discoverContextAvailabilityResponse</code> .
Availability subscription resource	/subscribeContextAvailability	Generic subscription to context availability information. The expected request body is an instance of <code>subscribeContextAvailabilityRequest</code> ; the response body is an instance of <code>subscribeContextAvailabilityResponse</code> .
Availability subscription update resource	/updateContextAvailabilitySubscription	Generic update of context availability subscriptions. The expected request body is an instance of <code>updateContextAvailabilitySubscriptionRequest</code> ; the response body is an instance of <code>updateContextAvailabilitySubscriptionResponse</code> .
Availability subscription deletion resource	/unsubscribeContextAvailability	Generic deletion of context availability subscriptions. The expected request body is an instance of <code>unsubscribeContextAvailabilityRequest</code> ; the response body is an instance of <code>unsubscribeContextAvailabilityResponse</code> .

### 2.1.2 Convenience Operation Resources

The table below gives an overview of the resources for convenience operation and the effects of interacting with them via the standard HTTP verbs GET, PUT, POST, and DELETE. The table cells contain only rough descriptions; the details can be found in Chapter 3.

Resource	Base URI: http://{serverRoot}/NGSI9	HTTP verbs			
		GET	PUT	POST	DELETE
Individual context entity	/contextEntities/{EntityID}	Retrieve information on providers of any information about the context entity	-	Register a provider of information about the entity	-
Attribute container of individual context entity	/contextEntities/{EntityID}/attributes				
Attribute of individual context entity	/contextEntities/{EntityID}/attributes/{attributeName}	Retrieve information on providers of the attribute value	-	Register a provider of information about the attribute	-
Attribute domain of individual context entity	/contextEntities/{EntityID}/attributeDomains/{attributeDomainName}	Retrieve information on providers of information about attribute values from the domain	-	Register a provider of information about attributes from the domain	-
Context entity type	/contextEntityType/{typeName}	Retrieve information on providers of any information	-	Register a provider of information about	-

		about context entities of the type		context entitie of the type	
Attribute container of entity type	/contextEntityTypes/{typeName}/attributes	Same functionality as parent resource, but implementation is optional.			
Attribute of entity type	/contextEntityTypes/{typeName}/attributes/{attributeName}	Retrieve information on providers of values of this attribute of context entities of the type	-	Register a provider of information about this attribute of context entities of the type	-
Attribute domain of entity type	/contextEntityTypes/{typeName}/attributeDomains/{attributeDomainName}	Retrieve information on providers of attribute values belonging to the specific domain, where the entity is of the specific type	-	Register a provider of information about attributes belonging to the specific domain, where the entity is of the specific type	-
Availability subscription container	/contextAvailabilitySubscriptions	-	-	Create a new availability subscription	-
Availability subscription	/contextAvailabilitySubscriptions/{subscriptionID}	-	Update subscription	-	Cancel subscription

## 2.2 Resource on Context Consumer Component

This section describes the resource that has to be provided by the context consumer in order to receive notifications. All attempts to interact with it by other verbs than POST shall result in an HTTP error status 405; the server should then also include the 'Allow: POST' field in the response.

Resource	URI	HTTP verbs
		POST
Notify context resource	//{notificationURI}	Generic availability notification. The expected request body is an instance of <code>notifyContextAvailabilityRequest</code> ; the response body is an instance of <code>notifyContextAvailabilityResponse</code> .

## 3. Convenience Function Resources

In this chapter, the resources which enable convenience functionality are described in details. In the context of this document, convenience functions are simple and intuitive interaction methods that are offered for the sake of convenience for application developers. Such methods are redundant in the sense that after removing them the system would still satisfy all functional requirements.

In the resource tree there are a number of resources that represent entities, attributes, and sets of entities and attributes. More specifically, these are the resources

- **`http://{serverRoot}/NGSI9/contextEntities`**
- **`http://{serverRoot}/NGSI9/contextEntityTypes`**

and all their subresources. The general approach of these resources is that with a GET operation one can retrieve information about where information on the respective entity/attribute sets is available, and what the respective metadata is.

Furthermore, the availability of context information about the entity/attribute sets represented by the resources can be announced using POST. For example, using POST on **`http://{serverRoot}/NGSI9/contextEntityTypes/person`** one can indicate the availability of context information about entities having the type “person”. Note that the level of granularity is determined by the resource interacted with, so **`http://{serverRoot}/NGSI9/contextEntityTypes/person`** cannot be used to register context information on individual entities, but only to indicate generally that information about persons is available at a certain URI. The request body of POST operations on all these resources is given as follows.

### 3.0.0.1 registerProviderRequest

Part name	Part type	Optional	Description
Metadata	ContextMetadata [0..unbounded]	Yes	Metadata characterizing this registration.
Duration	xsd:duration	Yes	Availability period
ProvidingApplication	xsd:anyURI	No	URI identifying the application that provides the values of the context attributes for the target Context Entities.
RegistrationId	xsd:string	Yes	Registration identifier to update previous registrations

## 3.1 Resource: Individual Context Entity

The resource used is

**`http://{serverRoot}/NGSI9/contextEntities/{entityID}`**

Instances of this resource type are used for retrieving which information about a certain context entity is available. Furthermore, the availability of context information about this entity can be announced by interaction with this resource.

The {entityID} part of the resource URI must denote the ID of an individual entity. The usage of regular expressions in {entityID} is supported neither here nor for any subresource.

### 3.1.1 GET

Retrieve all availability information about the context entity represented by the resource (i.e., URIs of all applications that provide context information about the entity).

The response body is an instance of `DiscoverContextAvailabilityResponse`. Its `ContextRegistrationResponseList` must only contain context availability information related to the context entity that is determined by the {entityID} part of the resource URI.

### 3.1.2 PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET, POST’ field in the response.

### 3.1.3 POST

This operation registers the availability of context information about the entity identified by the {entityID} part of the resource URI. Note that this operation can be used both for new registrations and for updates of existing registrations; in the latter case the request message should include the `RegistrationId` field.

The request message is an instance of `registerProviderRequest`. The response message is an instance of `registerContextResponse`.

### 3.1.4 DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, POST' field in the response.

## 3.2 Resource: Attribute Container of Individual Context Entity

The resource used is

**`http://{serverRoot}/NGSI9/contextEntities/{entityID}/attributes.`**

The support of interactions with this resource is optional. If supported, then the set of allowed interactions with this resource and their effects SHALL be equivalent to the parent resource.

## 3.3 Resource: Attribute of Individual Context Entity

The resource used is

**`http://{serverRoot}/NGSI9/contextEntities/{entityID}/attributes/{attributeName}`**

Instances of this resource are used for retrieving if and where values of the attribute {attributeName} of the entity determined by {entityID} can be retrieved, and for announcing the availability of such information.

### 3.3.1 GET

Retrieve all availability information about the attribute represented by the resource, including the respective metadata. The response body is an instance of `DiscoverContextAvailabilityResponse`, where the same restrictions as for GET operations on context entity resources apply. Additionally, only availability information related to the attribute determined by the {attributeName} part of the resource URI must be contained in the response.

### 3.3.2 PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, POST' field in the response.

### 3.3.3 POST

This operation registers the availability of values of attribute {attributeName} of the entity determined by the {entityID} part of the resource URI. This operation can be used both for new registrations and for updates of existing registrations; in the latter case the request message should include the `registrationId` field.

The request message is an instance of `registerProviderRequest`. The response message is an instance of `registerContextResponse`.

### 3.3.4 DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, POST' field in the response.

## 3.4 Resource: Attribute Domain of Individual Context Entity.

The resource used is



**http://{serverRoot}/NGSI9/contextEntities/{EntityID}/attributeDomains/{attributeDomainName}**

Instances of this resource are used for querying the availability of information about the attributes in the domain represented by the resource, and for registering the availability of this kind of information.

### 3.4.1 GET

Retrieve all availability information about the attributes from the domain represented by the resource, including the respective metadata. The response body is an instance of `DiscoverContextAvailabilityResponse`, where the same restrictions as for GET operations on context entity resources apply. Additionally, only availability information related to the attributes in the domain determined by the {attributeDomainName} part of the resource URI must be contained in the response.

### 3.4.2 PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, POST' field in the response.

### 3.4.3 POST

This operation registers the availability of values of attributes in the domain {attributeDomainName} of the entity determined by the {entityID} part of the resource URI. This operation can be used both for new registrations and for updates of existing registrations; in the latter case the request message should include the `registrationId` field.

The request message is an instance of `registerProviderRequest`. The response message is an instance of `registerContextResponse`.

### 3.4.4 DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, POST' field in the response.

## 3.5 Resource: Context Entity Type

The resource used is

**http://{serverRoot}/NGSI9/contextEntityTypes/{typeName}**

Instances of this resource are used for retrieving availability information about context entities of the type represented by the resource and announce the availability of such information.

Note that in NGSI-9/10, type names have the type `xs:anyURI`. Therefore, the {typeName} part of the resource access URI is a URI by itself. The {typeName} part has to be encoded in the resource URI as defined in RFC 3986, which means for example that any occurrence of the '\' character is encoded as '%2F'.

### 3.5.1 GET

Retrieve availability of information about context entities of the type represented by this resource. The response body is an instance of `discoverContextAvailabilityResponse`, which only contains availability information related to entities having type {typeName}.

The response can contain both

- availability information related to individual entities of type {typeName}
- information about where information about entities of type {typeName} is available (without the specification of individual entities)

### 3.5.2 PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, POST' field in the response.

### 3.5.3 POST

Register availability of information about context entities of the type represented by this resource. This operation is not intended to announce the availability of information about specific entities, but rather to indicate in general where information about entities of a type {typeName} is available. The operation can be used both for new registrations and for updates of existing registrations; in the latter case the request message should include the `registrationId` field.

The request message is an instance of `registerProviderRequest`. The response message is an instance of `registerContextResponse`.

### 3.5.4 DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, POST' field in the response.

## 3.6 Resource: Attribute Container of Context Entity Type

The resource used is

**`http://{serverRoot}/NGSI9/contextEntityTypes/{typeName}/attributes`.**

The support of interactions with this resource is optional. If supported, then the set of allowed interactions with this resource and their effects SHALL be equivalent to the parent resource.

## 3.7 Resource: Attribute of Context Entity Type

The resource used is

**`http://{serverRoot}/NGSI9/contextEntityTypes/{typeName}/attributes/{attributeName}`**

Instances of this resource are used for retrieving availability information regarding a specific attribute of all context entities having a certain type, and for announcing the availability of such kind of information.

### 3.7.1 GET

Retrieve information on where values of the attribute represented by the resource is available, and what the respective metadata is. The context entities of interest are all context entities of the type represented by the {typeName} part of the access URI.

The response body is an instance of `DiscoverContextAvailabilityResponse`, where the same restrictions as for GET operations on the parent resource apply. Additionally, only availability information related to the attribute determined by the {attributeName} part of the resource URI must be contained in the response.

### 3.7.2 PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, POST' field in the response.

### 3.7.3 POST

Register availability of values of attribute {attributeName} of context entities having the type represented by the {typeName} part of the access URI. The operation can be used both for new registrations and for updates of existing registrations; in the latter case the request message should include the `registrationId` field.

The request message is an instance of `registerProviderRequest`. The response message is an instance of `registerContextResponse`.

### 3.7.4 DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, POST' field in the response.

## 3.8 Resource: Attribute Domain of Context Entity Type

The resource used is

**`http://{serverRoot}/NGSI9/contextEntityTypes/{typeName}/attributeDomains/{attributeDomainName}`**

Instances of this resource are used for retrieving availability information regarding a specific attribute domain of all context entities having a certain type, and for announcing the availability of such kind of information.

### 3.8.1 GET

Retrieve information on where values of attributes from the domain represented by the resource are available, and what the respective metadata is. The context entities of interest are all context entities of the type represented by the {typeName} part of the access URI.

The response body is an instance of `DiscoverContextAvailabilityResponse`, where the same restrictions as for GET operations on the parent resource apply. Additionally, only availability information related to the attributes from the domain determined by the {attributeDomainName} part of the resource URI must be contained in the response.

### 3.8.2 PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, POST' field in the response.

### 3.8.3 POST

Register availability of values of attributes from the domain {attributeDomainName} of context entities having the type represented by the {typeName} part of the access URI. The operation can be used both for new registrations and for updates of existing registrations; in the latter case the request message should include the `registrationId` field.

The request message is an instance of `registerProviderRequest`. The response message is an instance of `registerContextResponse`.

### 3.8.4 DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, POST' field in the response.

## 3.9 Resource: Availability Subscription Container

The resource used is

**`http://{serverRoot}/NGSI9/contextAvailabilitySubscriptions`**

This is a container resource for context availability subscriptions, where new subscriptions are created via POST. Other operations SHOULD be rejected, as the modification and deletion of subscriptions is done by interaction with the corresponding child resources.

### 3.9.1 GET

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: POST' field in the response.

### 3.9.2 PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: POST' field in the response.

### 3.9.3 POST

Create a new context availability subscription. The type of the request and response body is `SubscribeContextAvailabilityRequest` and `SubscribeContextAvailabilityResponse`, respectively. The subscription will be represented by a new child resource, whose URI is determined by the subscription ID inside the response body. The recipient URI of the subscription is to be specified in the `Reference` field of the request body.

### 3.9.4 DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: POST' field in the response.

## 3.10 Resource: Availability Subscription

The resource used is

**`http://{serverRoot}/NGSI9/contextAvailabilitySubscriptions/{subscriptionID}`**

Instances of this resource represent context availability subscriptions. They can be interacted with via PUT and DELETE. Creation of new availability subscriptions is done by interaction with the parent resource.

### 3.10.1 GET

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: PUT, DELETE' field in the response.

### 3.10.2 PUT

This operation updates a context availability subscription. The request and response body is `updateContextAvailabilitySubscriptionRequest` and `updateContextAvailabilitySubscriptionResponse`, respectively. The `subscriptionId` field inside the request message must equal the `{subscriptionID}` part of the access URI, otherwise an error shall be returned.

### 3.10.3 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: PUT, DELETE' field in the response.

### 3.10.4 DELETE

This operation is used for deleting a context availability subscription, which corresponds to an `unsubscribeContextAvailability` operation. The request contains no body, and the response body is of type `UnsubscribeContextAvailabilityResponse`.

## 4. Data structure overview

The datastructures used in the RESTful binding defined in this document consist of following data structures:

- All data structures defined in the NGSI-9/10 specification.
- All input and output messages of NGSI-9
- A number of additional data structures that are defined in the subsections of this chapter. These additional structures are only used by the convenience functions, and their purpose is to avoid redundancy of information.

### registerProviderRequest

Part name	Part type	Optional	Description
Metadata	ContextMetadata [0..unbounded]	Yes	Metadata characterizing this registration.
Duration	xsd:duration	Yes	Availability period
ProvidingApplication	xsd:anyURI	No	URI identifying the application that provides the values of the context attributes for the target Context Entities.
RegistrationId	xsd:string	Yes	Registration identifier to update previous registrations