RESTful Network API for Communication Patterns

Candidate Version 1.0 – 12 May 2016

Open Mobile Alliance

OMA-TS-REST_NetAPI_CommunicationPatterns-V1_0-20160512-C

Contents

Figures

Tables

1. Scope

This specification defines a RESTful API for Communication Patterns using HTTP protocol bindings.

2. References

2.1 Normative References

[3GPP TS 23.682] 3GPP TS 23.682 V13.4.0 "Technical Specification Group Services and System Aspects;

Architecture enhancements to facilitate communications with packet data networks and applications (Release 13)", 3rd Generation Partnership Project, December 2015, URL:http://www.3gpp.org/

[Autho4API_10] "Authorization Framework for Network APIs", Open Mobile Alliance™, OMA-ER-Autho4API-V1_0, URL:http://www.openmobilealliance.org/

[ENCap_M2M] "Exposing Network Capabilities to M2M Requirements", Open Mobile Alliance™, OMA-RD-ENCap_M-V1_0, URL:http://www.openmobilealliance.org/

[REST_NetAPI_ACR] "RESTful Network API for Anonymous Customer Reference Management ", Open Mobile Alliance™, OMA-TS-REST_NetAPI_ACR-V1_0, URL:http://www.openmobilealliance.org/

[REST_NetAPI_Common] "Common definitions for RESTful Network APIs", Open Mobile Alliance™, OMA-TS-REST_NetAPI_Common-V1_0, URL:http://www.openmobilealliance.org/

[REST_NetAPI_Common] "Common definitions for RESTful Network APIs", Open Mobile Alliance™, OMA-TS-REST_NetAPI_Common-V1_0, URL:http://www.openmobilealliance.org/

[REST_NetAPI_NotificationChannel] "RESTful Network API for Notification Channel", Open Mobile Alliance™, OMA-TS-REST_NetAPI_NotificationChannel-V1_0, URL:http://www.openmobilealliance.org/

[REST_SUP_COMPATT] "XML schema for the RESTful Network API for Communication Patterns", Open Mobile Alliance™, OMA-SUP-XSD_rest_netapi_communicationpatterns-V1_0, URL:http://www.openmobilealliance.org/

[RFC2119] "Key words for use in RFCs to Indicate Requirement Levels", S. Bradner, March 1997, URL:http://tools.ietf.org/html/rfc2119.txt

[RFC3966] "The tel URI for Telephone Numbers", H.Schulzrinne, December 2004, URL:http://tools.ietf.org/html/rfc3966.txt

[RFC3986] "Uniform Resource Identifier (URI): Generic Syntax", R. Fielding et. al, January 2005, URL:http://tools.ietf.org/html/rfc3986.txt

[RFC5777] "Traffic Classification and Quality of Service (QoS) Attributes for Diameter", J. Korhonen et. al, February 2010, URL:http://tools.ietf.org/html/rfc5777.txt

[RFC7159] "The JavaScript Object Notation (JSON) Data Interchange Format", T. Bray, Ed., March 2014, URL:http://tools.ietf.org/html/rfc7159.txt

[RFC7231] Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content, R. Fielding, Ed., J.Raschke, Ed., June 2014, URL:http://tools.ietf.org/html/rfc7231.txt

[SCRRULES] "SCR Rules and Procedures", Open Mobile Alliance™, OMA-ORG-SCR_Rules_and_Procedures, URL:http://www.openmobilealliance.org/

[XMLSchema1] W3C XML Schema Definition Language (XSD) 1.1 Part 1: Structures, W3C Recommendation 5 April 2012, URL:http://www.w3.org/TR/xmlschema11-1/

[XMLSchema2] W3C XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes, W3C Recommendation 5 April 2012, URL:http://www.w3.org/TR/xmlschema11-2/

2.2 Informative References

[OMADICT] "Dictionary for OMA Specifications", Version 2.9, Open Mobile Alliance™, OMA-ORG-Dictionary-V2_9, URL:http://www.openmobilealliance.org/

[REST_WP] "Guidelines for RESTful Network APIs", Open Mobile Alliance™, OMA-WP-Guidelines_for_RESTful_Network_APIs, URL:http://www.openmobilealliance.org/

3. Terminology and Conventions

3.1 Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

All sections and appendixes, except “Scope” and “Introduction”, are normative, unless they are explicitly indicated to be informative.

3.2 Definitions

For the purpose of this TS, all definitions from the OMA Dictionary apply [OMADICT].

Client-side Notification URL
An HTTP URL exposed by a client, on which it is capable of receiving notifications and that can be used by the client when subscribing to notifications.

Heavy-weight Resource
A resource which is identified by a resource URL which is then used by HTTP methods to operate on the entire data structure representing the resource. Include this definition if Light-weight Resources are supported, otherwise delete it.

Light-weight Resource
A subordinate resource of a Heavy-weight Resource which is identified by its own resource URL which is then used by HTTP methods to operate on a part of the data structure representing the Heavy-weight Resource. The Light-weight Resource URL can be seen as an extension of the Heavy-weight Resource URL.

There could be several levels of Light-weight Resources below the ancestor Heavy-weight Resource, depending on the data structure. Include this definition if Light-weight Resources are supported, otherwise delete it.

Long Polling
A variation of the traditional polling technique, where the server does not reply to a request unless a particular event, status or timeout has occurred. Once the server has sent a response, it closes the connection, and typically the client immediately sends a new request. This allows the emulation of an information push from a server to a client.

Notification Channel
A channel created on the request of the client and used to deliver notifications from a server to a client. The channel is represented as a resource and provides means for the server to post notifications and for the client to receive them via specified delivery mechanisms.

For example in the case of Long Polling the channel resource is defined by a pair of URLs. One of the URLs is used by the client as a call-back URL when subscribing for notifications. The other URL is used by the client to retrieve notifications from the Notification Server.

Notification Server
A server that is capable of creating and maintaining Notification Channels.

Server-side Notification URL
An HTTP URL exposed by a Notification Server, that identifies a Notification Channel and that can be used by a client when subscribing to notifications.

3.3 Abbreviations

ACR	Anonymous Customer Reference
API	Application Programming Interface
HTTP	HyperText Transfer Protocol
JSON	JavaScript Object Notation
MIME	Multipurpose Internet Mail Extensions
OMA	Open Mobile Alliance
REST	REpresentational State Transfer
SCR	Static Conformance Requirements
SIP	Session Initiation Protocol

TS	Technical Specification
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
WP	White Paper
XML	eXtensible Markup Language
XSD	XML Schema Definition

4. Introduction

The Technical Specification of the RESTful Network API for Communication Patterns contains HTTP protocol bindings for providing information about predictable communication patterns of individual Devices or groups of Devices, using the REST architectural style. The specification provides resource definitions, the HTTP verbs applicable for each of these resources, and the element data structures, as well as support material including flow diagrams and examples using the various supported message body formats (i.e. XML, JSON).

The RESTful Network API for Communication Patterns enables a third party/M2M application to provide the network operator with information about predictable communication patterns (CP) of individual devices or a group of devices that are served by this network operator. This information may be used by the Network Operator to optimize resource usage.

In M2M use cases, a third party can often predict kinds of communication patterns of individual Devices or a group of Devices and may be configured to have specific communication patterns depending on application data. The application data here are, for example, temperature observed by a thermometer, telemetry data observed by an Engine Control Unit (ECU), water levels observed by a flood level sensor, and so on. See the detailed use cases in OMA Exposing Network Capabilities to M2M [ENCap_M2M].

It's hard for a Network Operator to make such a prediction since it should be agnostic on application data. The RESTful Network API for Communication Patterns facilitates the interaction on information about predictable communication patterns between a third party and a Network Operator in a standardized manner.

4.1 Version 1.0

The RESTful Network API for Communication Patterns V1.0 is a part of the suite of OMA RESTful Network APIs. The requirements for this API are found in OMA Exposing Network Capabilities to M2M RD [OMA ENCap-M2M].

Version 1.0 of this specification supports the following operations:

- Submitting information about predictable communication patterns of Devices (creating communication patterns resource)
- Retrieving a list of submitted information about communication patterns
- Retrieving submitted information about an individual communication pattern
- Updating submitted information about individual communication pattern
- Deleting submitted information about individual communication pattern (deleting the communication pattern resource)
- Retrieving the configuration status of network infrastructure in response to submitted information about communication patterns

In addition this specification provides:

- Support for scope values used with authorization framework defined in [Autho4API_10]
- Support for Anonymous Customer Reference (ACR) as an end user identifier
- Support for "acr:auth" as a reserved keyword in an ACR

In version 1.0 of this specification, predictable communication patterns (CP) include time patterns, location and mobility related patterns as specified in OMA Exposing Network Capabilities to M2M [ENCap_M2M]. These patterns are in line with 3GPP TS 23.682 [3GPP TS 23.682].

In 3GPP Rel-13, predictable communication patterns of the devices are provided through Service Capability Exposure Function (SCEF) to network entities appropriately to optimise network resources for those devices' communications.

5. Communication Patterns API definition

This section is organized to support a comprehensive understanding of the Communication Patterns API design. It specifies the definition of all resources, definition of all data structures, and definitions of all operations permitted on the specified resources.

Common data types, naming conventions, fault definitions and namespaces are defined in [REST_NetAPI_Common].

The remainder of this document is structured as follows:

Section 5 starts with a diagram representing the resources hierarchy followed by a table listing all the resources (and their URL) used by this API, along with the data structure and the supported HTTP verbs (section 5.1). What follows are the data structures (section 5.2). A sample of typical use cases is included in section 5.3, described as high level flow diagrams.

Section 6 contains detailed specification for each of the resources. Each such subsection defines the resource, the request URL variables that are common for all HTTP methods, and the supported HTTP verbs. For each supported HTTP verb, a description of the functionality is provided, along with an example of a request and an example of a response. For each unsupported HTTP verb, the returned HTTP error status is specified, as well as what should be returned in the Allow header.

All examples in section 6 use XML as the format for the message body. JSON examples are provided in Appendix C.

Section 7 contains fault definition details such as Service Exceptions and Policy Exceptions.

Appendix B provides the Static Conformance Requirements (SCR).

Appendix D provides the operations mapping to a pre-existing baseline specification, where applicable.

Appendix E provides a list of all Light-weight Resources, where applicable.

Appendix F defines authorization aspects to control access to the resources defined in this specification.

Note: Throughout this document client and application can be used interchangeably.

5.1 Resources Summary

This section summarizes all the resources used by the RESTful Network API for Communication Patterns.

The "apiVersion" URL variable SHALL have the value "v1" to indicate that the API corresponds to this version of the specification. See [REST_NetAPI_Common] which specifies the semantics of this variable.

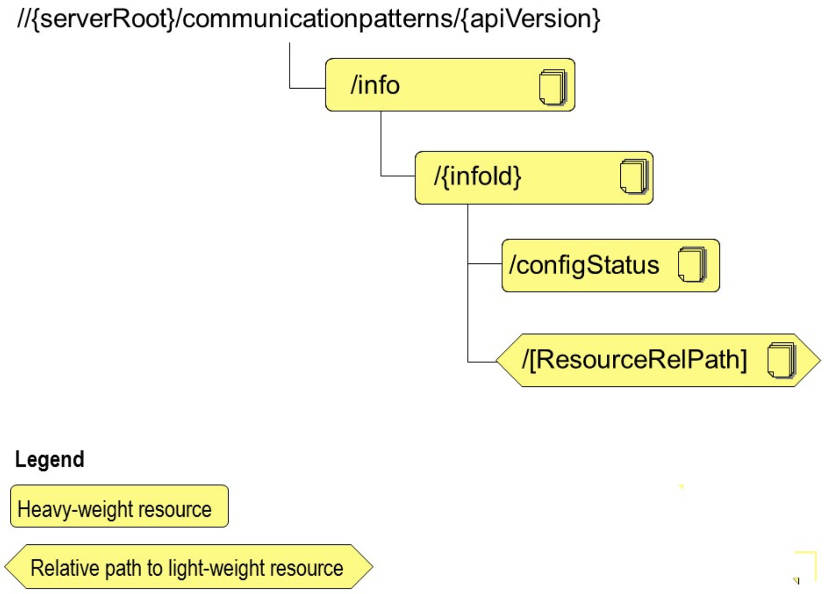
//{serverRoot}/communicationpatterns/{apiVersion}
/info
/{infoId}
/configStatus
/[ResourceRelPath]
Legend
Heavy-weight resource
Relative path to light-weight resource

Figure 1 Resource structure defined by this specification

The following tables give a detailed overview of the resources defined in this specification, the data type of their representation and the allowed HTTP methods.

Purpose: To allow client to submit and manage information about predictable communication patterns of devices and obtain the configuration status

Resource	URL Base URL: http://{serverRoot}/ communicationpatterns{apiVersion }	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
Communication patterns information	/info	InfoList (used for GET) Info (used for POST) common:Resource Reference (optional alternative for POST response)	Retrieve a list of communication patterns information	no	Submit a new Communication pattern information	no
Communication patterns information instance	/info/{infoId}	Info (used for GET and PUT)	Retrieve an individual Communication pattern information	Update an individual Communication pattern information	no	Delete an individual Communication pattern information
Configuration status	/info/{infoId}/configStatus	ConfigStatus	Retrieve the configuration status	no	no	no
Individual subset of Information about predictable communication patterns of Devices	/info/{infoId}/[RelativeRelPath]	The data structure corresponds to an element within Info pointed out by request URL	no	Update subset of Info (Information about predictable communication patterns of Devices)	no	no

5.2 Data Types

5.2.1 XML Namespaces

The XML namespace for the Communication Patterns data types is:

urn:oma:xml:rest:communicationpatterns:1

The 'xsd' namespace prefix is used in the present document to refer to the XML Schema data types defined in XML Schema [XMLSchema1, XMLSchema2]. The 'common' namespace prefix is used in the present document to refer to the data types defined in [REST_NetAPI_Common]. The use of namespace prefixes such as 'xsd' is not semantically significant.

The XML schema for the data structures defined in the section below is given in [REST_SUP_CommunicationPatterns].

5.2.2 Structures

The subsections of this section define the data structures used in the Communication Patterns API.

Some of the structures can be instantiated as so-called root elements.

For structures that contain elements which describe a user identifier, the statements in section 6 regarding 'tel', 'sip' and 'acr' URI schemes apply.

5.2.2.1 Type: InfoList

A type containing list of submitted information about predictable communication patterns of devices

Element	Type	Optional	Description
info	Info [0..unbounded]	Choice	Information about predictable communication patterns of Devices
infoReference	common:ResourceReference [0..unbounded]	Choice	Reference to a resource which represents a submitted information (i.e. {infoId})
resourceURL	xsd:anyURI	Yes	Self referring URL.

A root element named infoList of type InfoList is allowed in response bodies.

XSD modelling uses a “choice” to select either info or infoReference, but not both of them.

5.2.2.2 Type: Info

A type containing information about predictable communication patterns of devices.

Element	Type	Optional	[ResourceRelPath]	Description
serial	xsd:string	No	Not Applicable	An identifier or a tag provided by the client application for this resource.
source	xsd:anyURI	Yes	Not applicable	Address of the information source. Note the application may predict communication patterns of Devices by itself, or may outsource the prediction.
target	TargetDevices	No	target	List of all the devices which are predicted to have this communication patterns

validity	Validity	No	validity	Valid start and end time of this communication pattern information
parameterSet	ParameterSet [1..unbounded]	No	parameterSet	Parameter set of communication patterns
resourceURL	xsd:anyURI	Yes	Not applicable	Self referring URL. The resourceURL SHALL NOT be included in POST requests by the client, but MUST be included in POST requests representing notifications by the server to the client, when a complete representation of the resource is embedded in the notification. The resourceURL MUST be also included in responses to any HTTP method that returns an entity body, and in PUT requests.

A root element named info of type Info is allowed in request and/or response bodies.

5.2.2.3 Type: TargetDevices

A type containing the list of submitted information about predictable communication patterns is applicable to.

Element	Type	Optional	Description
address	xsd:anyURI [1..unbounded]	No	Address of the Device which is predicted to have the communication patterns (e.g., 'sip' URI, 'tel' URI, 'acr' URI). Reference to a group could be provided here if supported by implementation.

5.2.2.4 Type: ParameterSet

A type containing the parameter set of the predictable communication patterns.

Element	Type	Optional	Description
label	xsd:string	No	An identifier or a tag provided by the client application for a parameter set.
periodicCommmunication	PeriodicCommmunication	No	Identifies if the application communicates periodically or not. And if communication is periodic then duration and interval of periodicity is defined.
scheduledCommunication	TimeRange [0..unbounded]	Yes	Time zone and Day of the week when the application is available for communication
stationary	Stationary	No	Identifies the position of the application, e.g. static or mobile

5.2.2.5 Type: PeriodicCommmunication

A type containing periodic communication indicator.

Attribute	Type	Optional	Description
commIndicator	CommIndicator	No	Indicates the nature of communication.

Element	Type	Optional	Description
duration	xsd:unsignedInt	Yes	Defines the communication duration time in seconds. Valid when indicator="0".
interval	xsd:unsignedInt	Yes	Defines the frequency of communication in seconds (for example 3600 for one hour interval). Valid when indicator="0"

5.2.2.6 Type: TimeRange

A type containing time zone and day of the week when the application is available for communication (for schedule communication).

Element	Type	Optional	Description
timeOfDayStart	xsd:unsignedInt	No	Time-Of-Day-Condition, IETF RFC 5777 [RFC5777]. For example 0 is 0.00am
timeOfDayEnd	xsd:unsignedInt	No	Time-Of-Day-Condition, as defined in IETF RFC 5777 [RFC5777]. For example 18000 is 5.00am
dayOfWeekMask	xsd:unsignedInt	No	Time-Of-Day-Condition, as defined in IETF RFC 5777 [RFC5777].For example 1111100 is SAT FRI THU WED TUE
timezoneFlag	xsd:unsignedInt	No	Time-Of-Day-Condition, as defined in IETF RFC 5777 [RFC5777]. For example 0: UTC, 1: LOCAL, 2: OFFSET

5.2.2.7 Type: Stationary

A type containing mobility nature of devices for the purposes of communication pattern parameter set.

Attribute	Type	Optional	Description
mobilityIndicator	MobilityIndicator	No	Indicates if the device is stationary or mobile.

Element	Type	Optional	Description
trajectory	xsd:CDATA	Yes	Operator specific or vendor specific expression of a predictable trajectory of Devices, e.g. GPX or KML

5.2.2.8 Type: Status

A type containing configuration status of communication patterns parameter set.

Element	Type	Optional	Description
address	xsd:anyURI	No	URI (for example external identifier or MSISDN) of a single device
code	StatusCode	No	Configuration status for submitted communication patterns information

5.2.2.9 Type: Validity

A type containing a period that information about predictable communication patterns is valid for.

Element	Type	Optional	Description
startTime	xsd:dateTime	Yes	If present defines the start time of validity of a particular communication pattern.
expiryTime	xsd:dateTime	No	It defines the end time of validity of a particular communication pattern.

5.2.2.10 Type: ConfigStatus

A type containing configuration status of communication patterns parameter set.

Element	Type	Optional	Description
status	Status [1..unbounded]	No	List of configuration status of communication patterns parameter set
resourceURL	xsd:anyURI	No	Self referring URL.
link	common:Link	Yes	Link reference for the resource representing theinformation (i.e., {infoId})

5.2.3 Enumerations

The subsections of this section define the enumerations used in the Communication Patterns API.

5.2.3.1 Enumeration: StatusCode

List of possible values for configuration status for submitted communication patterns information.

Enumeration	Description
UpdateRequestQueued	This indicates that communication patterns parameterset is queued and not sent to the network yet
UpdateRequestSent	This indicates that network configuration related to a communication patterns parameterset is in progress
Updated	Final state that indicates that the network provisioning of the communication patterns parameter set(s) was successful
UpdateImpossible	This indicates that the network provisioning of the communication patterns parameter set(s) failed
UpdateUnknown	This indicates that configuration status related to a communication patterns parameterset is unknown e.g, no response from network
OutOfValidityTime	Communication patterns information is not within the validity time.

5.2.3.2 Enumeration: CommIndicator

Enumeration	Description
OnDemand	This indicates that communication style of the device(s) is on demand in nature.
Periodic	This indicates that communication style of the device(s) is periodic in nature.

5.2.3.3 Enumeration: MobilityIndicator

Enumeration	Description
Stationary	This indicates that the device is not is motion.
Mobile	This indicates that the device is mobile.

5.2.4 Values of the Link “rel” attribute

The “rel” attribute of the Link element is a free string set by the server implementation, to indicate a relationship between the current resource and an external resource. The following are possible strings (list is non-exhaustive, and can be extended):

- InfoReference

These values indicate the kind of resource that the link points to.

5.3 Sequence Diagrams

The following subsections describe the resources, methods and steps involved in typical scenarios.

In a sequence diagram, a step which involves delivering a notification is labeled with “POST or NOTIFY”, where “POST” refers to delivery via the HTTP POST method, and “NOTIFY” refers to delivery using the Notification Channel [REST_NetAPI_NotificationChannel].

5.3.1 Submit information about predictable communication patterns and check the configuration status

This figure below shows a scenario for submitting information about predictable communication patterns of a device or a group of devices.

The resources:

- To submit information about predictable communication patterns, create resource under **http://{serverRoot}/communicationpatterns/{apiVersion}/info**
- To retrieve the configuration status in regard to the submitted information, get resource of **http://{serverRoot}/communicationpatterns/{apiVersion}/info/{infoId}/configStatus**

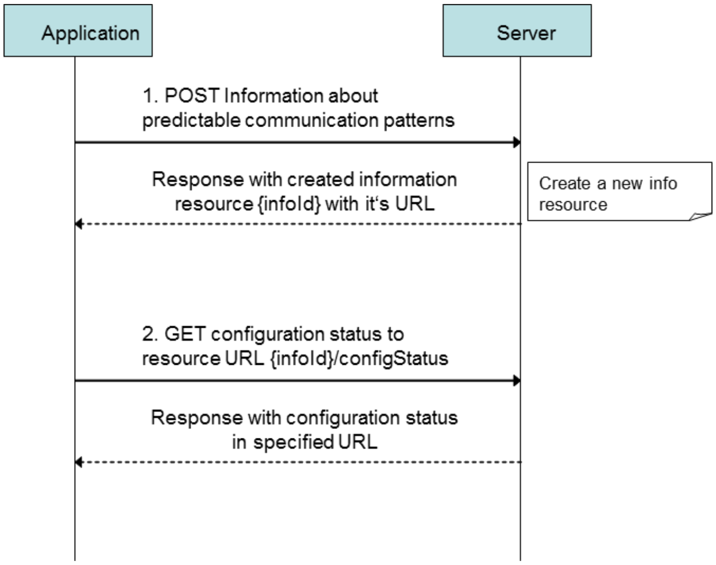
Application
Server
1. POST Information about predictable communication patterns
Response with created information resource {infoId} with it's URL
Create a new info resource
2. GET configuration status to resource URL {infoId}/configStatus
Response with configuration status in specified URL

Figure 2 Submission of new information and checking the status

Outline of the flows:

1. An application initiates the creation of new info resource for predictable communication pattern using POST and receives the created resource with a resource URL containing the infoId
2. The application requests the configuration status in resource URL {infoId}/configStatus using GET and receives the response with configuration status in specified URL.

5.3.2 Update the information about predictable communication patterns

This figure below shows the call flow for updating information about communication patterns of a device or a group of devices.

The resources:

- To retrieve the communication pattern information of a particular infoId, get resource of **http://{serverRoot}/communicationpatterns/{apiVersion}/info/{infoId}**
- To update the information of an instance of communication patterns, put resource under **http://{serverRoot}/communicationpatterns/{apiVersion}/info/{infoId}**

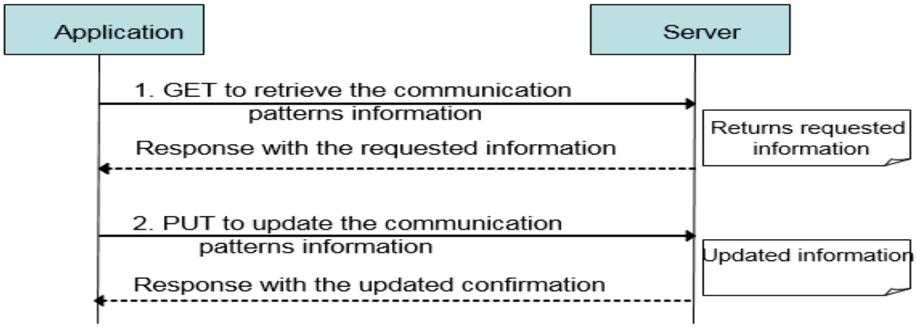
Application
Server
1. GET to retrieve the communication patterns information
Returns requested information
Response with the requested information
2. PUT to update the communication patterns information
Updated information
Response with the updated confirmation

Figure 3 Flow for updating existing communication pattern parameters

Outline of the flows:

1. An application requests the retrieval of communication pattern information using GET for a particular infoId and the server responses with the communication pattern information/parameters.
2. Application submits new communication pattern parameters/information by using PUT method to update the infoId and server responses with the updated communication pattern information.

5.3.3 Query about submitted communication patterns list and configuration status

This figure below shows the call flow for querying about submitted communication patterns information and for retrieving configuration status.

The resources:

- To query about submitted communication patterns list, get resource under **http://{serverRoot}/communicationpatterns/{apiVersion}/info**
- To retrieve the configuration status of communication pattern information of a particular infoId, get resource of **http://{serverRoot}/communicationpatterns/{apiVersion}/info/{infoId}/configStatus**

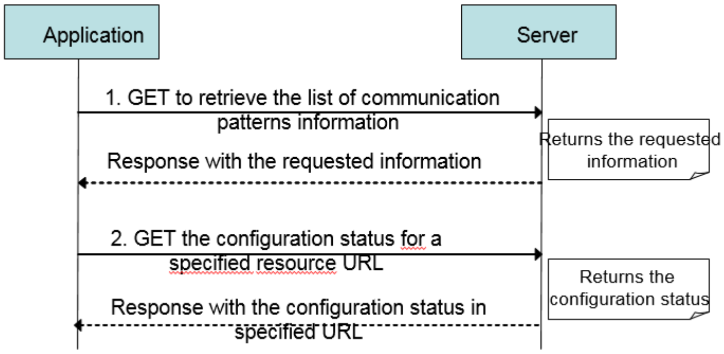
Application
Server
1. GET to retrieve the list of communication patterns information
Returns the requested information
Response with the requested information
2. GET the configuration status for a specified resource URL
Returns the configuration status
Response with the configuration status in specified URL

Figure 4 Flow for querying the configuration status

Outline of the flows:

1. An application queries about communication patterns information list using GET on the info resource and server returns communication patterns information list.
2. An application then initiates the retrieval of the configuration status using GET for a particular infoId and the server returns the configuration status of the specified resource.

5.3.4 Deletion of communication patterns information

This figure below shows the call flow for deleting information about communication patterns of a device or group of devices.

The resources:

- To query information about submitted communication patterns list, get resource of **http://{serverRoot}/communicationpatterns/{apiVersion}/info**
- To delete the information about an instance of communication patterns, remove resource of **http://{serverRoot}/communicationpatterns/{apiVersion}/info/{infoId}**

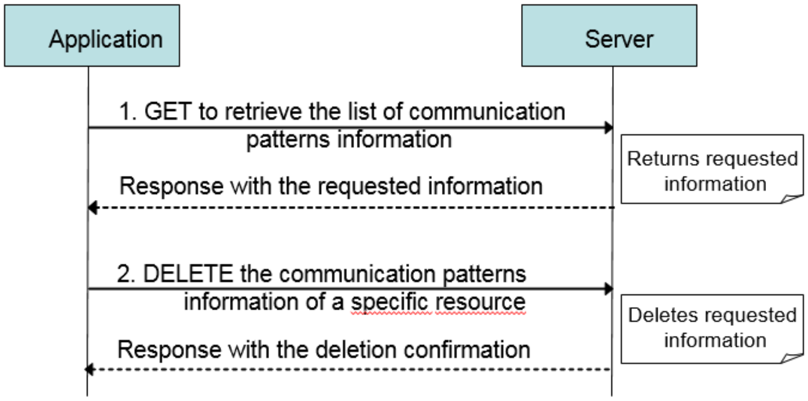
Application
Server
1. GET to retrieve the list of communication patterns information
Returns requested information
Response with the requested information
2. DELETE the communication patterns information of a specific resource
Deletes requested information
Response with the deletion confirmation

Figure 5 Flow for deleting a broadcast message

Outline of the flows:

1. An application queries about communication patterns information list using GET on info resource and the server responses with the list of submitted communication pattern information.
2. The application decides to remove one of the submitted communication pattern information in the list by using DELETE method on the resource infoId and the server returns a response confirming deletion.

5.3.5 Query about submitted communication patterns and updating individual communication patterns information subset

This figure below shows the call flow for updating information about communication patterns of a device or a group of devices.

The resources:

- To retrieve the communication pattern information of a particular infoId, get resource of **http://{serverRoot}/communicationpatterns/{apiVersion}/info/{infoId}**
- To update the information of an instance of communication patterns, put resource under **http://{serverRoot}/communicationpatterns/{apiVersion}/info/{infoId}**
- To update individual information (subset) of an instance of communication patterns, put resource under the following Light-weight Resource **http://{serverRoot}/communicationpatterns/{apiVersion}/info/{infoId}/[ResourceRelPath]**

Where [ResourceRelPath] is a light-weight relative resource URL, and it should be replaced with a string corresponding to the parameter that needs to be updated. Those strings are defined in [ResourceRelPath] column in the tables in section 5.2.2.

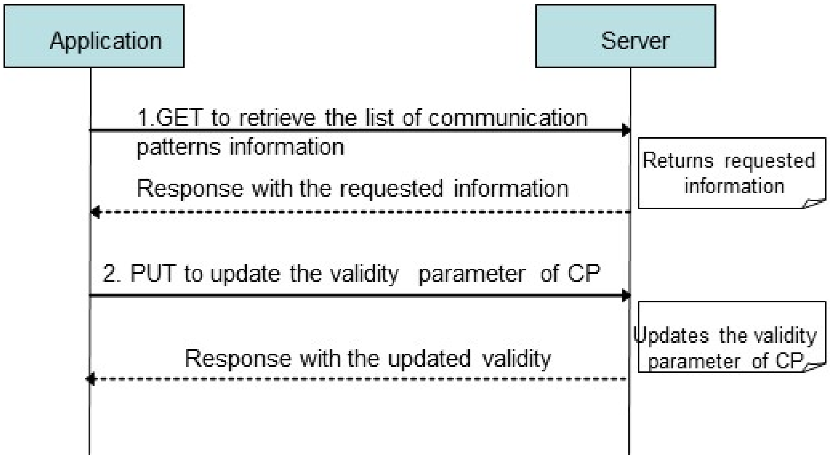
Application
Server
1.GET to retrieve the list of communication patterns information
Returns requested information
Response with the requested information
2. PUT to update the validity parameter of CP
Updates the validity parameter of CP
Response with the updated validity

Figure 6 Flow for updating validity parameter of communication patterns information

Outline of the flows:

1. An application requests the retrieval of communication pattern information using GET for a particular infoId and the server responses with the communication pattern information/parameters.
2. Application decides to change the validity parameter that was initially submitted for the of communication patterns information by using PUT method on the resource identified by the resource URL. Application is using PUT method on a Light-weight Resource for updating validity parameter. In this case URL parameter [ResourceRelPath] shall be replaced with a string “validity”. The server returns an HTTP response confirming updated validity parameter.

6. Detailed specification of the resources

The following applies to all resources defined in this specification regardless of the representation format (i.e. XML, JSON):

- Reserved characters in URL variables (parts of a URL denoted below by a name in curly brackets) MUST be percent-encoded according to [RFC3986]. Note that this always applies, no matter whether the URL is used as a Request URL or inside the representation of a resource (such as in “resourceURL” and “link” elements).
- If a user identifier (e.g. address, participantAddress, etc.) of type anyURI is in the form of an MSISDN, it MUST be defined as a global number according to [RFC3966] (e.g. tel:+19585550100). The use of characters other than digits and the leading “+” sign SHOULD be avoided in order to ensure uniqueness of the resource URL. This applies regardless of whether the user identifier appears in a URL variable or in a parameter in the body of an HTTP message.
- If an equipment identifier of type anyURI is in the form of a SIP URI, it MUST be defined according to [RFC3261].
- If a user identifier (e.g. address, userId, etc) of type anyURI is in the form of an Anonymous Customer Reference (ACR), it MUST be defined according to [REST_NetAPI_ACR], i.e. it MUST include the protocol prefix 'acr:' followed by the ACR.
    - o The ACR ‘auth’ is a supported reserved keyword, and MUST NOT be assigned as an ACR to any particular end user. See F.1.2 for details regarding the use of this reserved keyword.
- For requests and responses that have a body, the following applies: in the requests received, the server SHALL support JSON and XML encoding of the parameters in the body. The Server SHALL return either JSON or XML encoded parameters in the response body, according to the result of the content type negotiation as specified in [REST_NetAPI_Common]. In notifications to the Client, the server SHALL use either XML or JSON encoding, depending on which format the client has specified in the related subscription. The generation and handling of the JSON representations SHALL follow the rules for JSON encoding in HTTP Requests/Responses as specified in [REST_NetAPI_Common].

6.1 Resource: Communication patterns information

The resource used is:

http://{serverRoot}/communicationpatterns/{apiVersion}/info

This resource is used for retrieving an already issued communication patterns information list and for creating new infoId for submitting communication patterns information.

6.1.1 Request URL variables

The following request URL variables are common for all HTTP methods:

Name	Description
serverRoot	Server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: example.com/exampleAPI
apiVersion	Version of the API client wants to use. The value of this variable is defined in section 5.1

See section 6 for a statement on the escaping of reserved characters in URL variables.

6.1.2 Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common].

For Policy Exception and Service Exception fault codes applicable to Communication Patterns API, see section 7.

6.1.3 GET

This operation is used for querying about submitted communication patterns information list.

6.1.3.1 Example 1: Retrieving a list of communication patterns information (Informative)

6.1.3.1.1 Request

```
GET /exampleAPI/communicationpatterns/v1/info HTTP/1.1
Host: example.com
Accept: application/xml
```

6.1.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Thu, 07 Mar 2016 11:00:00 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<cp:infoList xmlns:cp="urn:oma:xml:rest:netapi:communicationpatterns:1">
  <info>
    <serial>A00001EF</serial>
    <source>http://www.thirdparty.example.com</source>
    <target>
      <address>tel:+19585550100</address>
      <address>tel:+19585550101</address>
      <address>tel:+19585550102</address>
    </target>
    <validity>
      <startTime>2016-06-23T18:45:00-07:00</startTime>
      <expiryTime>2017-06-23T18:45:00-07:00</expiryTime>
    </validity>
    <parameterSet>
      <label>Request 1</label>
      <periodicCommunication commIndicator="Periodic">
        <duration>300</duration>
        <interval>7200</interval>
      </periodicCommunication>
      <scheduledCommunication>
                              <timeOfDayStart>0</timeOfDayStart>
                              <timeOfDayEnd>18000</timeOfDayEnd>
                              <dayOfWeekMask>1111100</dayOfWeekMask>
                              <timezoneFlag>1</timezoneFlag>
      </scheduledCommunication>
      <scheduledCommunication>
                              <timeOfDayStart>72000</timeOfDayStart>
                              <timeOfDayEnd>0</timeOfDayEnd>
                              <dayOfWeekMask>0111110</dayOfWeekMask>
                              <timezoneFlag>1</timezoneFlag>
      </scheduledCommunication>
      <stationary mobilityIndicator="Mobile">
        <trajectory>
        </trajectory>
```

```
   </stationary>
  </parameterSet>
  <resourceURL>http://example.com/exampleAPI/communicationpatterns/v1/info/i0001</resourceURL>
 </info>
 <info>
  <serial>A00001F0</serial>
  <source>http://www.thirdparty.example.com</source>
  <target>
   <address>tel:+19585550103</address>
   <address>tel:+19585550104</address>
  </target>
  <validity>
   <startTime>2016-06-23T09:00:00</startTime>
   <expiryTime>2017-06-23T09:00:00</expiryTime>
  </validity>
  <parameterSet>
   <label>Request 2</label>
   <periodicCommunication commIndicator="Periodic">
    <duration>300</duration>
    <interval>3600</interval>
   </periodicCommunication>
   <scheduledCommunication>
                 <timeOfDayStart>32400</timeOfDayStart>
                 <timeOfDayEnd>61200</timeOfDayEnd>
                 <dayOfWeekMask>1000001</dayOfWeekMask>
                 <timezoneFlag>1</timezoneFlag>
   </scheduledCommunication>
   <scheduledCommunication>
                 <timeOfDayStart>72000</timeOfDayStart>
                 <timeOfDayEnd>0</timeOfDayEnd>
                 <dayOfWeekMask>0111110</dayOfWeekMask>
                 <timezoneFlag>1</timezoneFlag>
   </scheduledCommunication>
   <stationary mobilityIndicator="Stationary">
    <trajectory>
    </trajectory>
   </stationary>
  </parameterSet>
  <resourceURL>http://example.com/exampleAPI/communicationpatterns/v1/info/i0002</resourceURL>
 </info>
<resourceURL>http://example.com/exampleAPI/communicationpatterns/v1/info</resourceURL>
</cp:infoList>
```

6.1.3.2 Example 2: Retrieving a list of communication patterns information, response with a list of resource references (Informative)

6.1.3.2.1 Request

```
GET /exampleAPI/communicationpatterns/v1/info HTTP/1.1
Host: example.com
Accept: application/xml
```

6.1.3.2.2 Response

```
HTTP/1.1 200 OK
Date: Thu, 07 Mar 2016 11:00:00 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<cp:infoList xmlns:cp="urn:oma:xml:rest:netapi:communicationpatterns:1">
  <infoReference>
    <resourceURL>http://example.com/exampleAPI/communicationpatterns/v1/info/i0001</resourceURL>
  </infoReference>
  <infoReference>
    <resourceURL>http://example.com/exampleAPI/communicationpatterns/v1/info/i0002</resourceURL>
  </infoReference>
</cp:infoList>
```

6.1.4 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 as per sections 6.5.5 and 7.4.1 of [RFC7231].

6.1.5 POST

This operation is used for creating new infoId for submitting communication patterns information.

6.1.5.1 Example 1: Submitting new information about predictable communication patterns of Devices (Informative)

In this example an application provides the server with new information about predictable communication patterns of Devices. When the application submits new information to the server, the server allocates a new {infoId} resource representing the information.

6.1.5.1.1 Request

```
POST /exampleAPI/communicationpatterns/v1/info HTTP/1.1
Host: example.com
Accept: application/xml
Content-Length: nnnn
MIME-Version: 1.0

<?xml version="1.0" encoding="UTF-8"?>
<cp:info xmlns:cp="urn:oma:xml:rest:netapi:communicationpatterns:1">
  <serial>89ABCDEF</serial>
  <source>http://www.thirdparty.example.com</source>
  <target>
    <address>tel:+19585550105</address>
    <address>tel:+19585550106</address>
    <address>tel:+19585550107</address>
  </target>
  <validity>
    <startTime>2016-06-23T09:00:00</startTime>
    <expiryTime>2017-06-23T09:00:00</expiryTime>
  </validity>
  <parameterSet>
    <label>Request 3</label>
    <periodicCommunication commIndicator="Periodic">
      <duration>300</duration>
      <interval>7200</interval>
    </periodicCommunication>
    <scheduledCommunication>
      <timeOfDayStart>0</timeOfDayStart>
      <timeOfDayEnd>18000</timeOfDayEnd>
      <dayOfWeekMask>1111100</dayOfWeekMask>
      <timezoneFlag>1</timezoneFlag>
    </scheduledCommunication>
    <scheduledCommunication>
      <timeOfDayStart>72000</timeOfDayStart>
      <timeOfDayEnd>0</timeOfDayEnd>
      <dayOfWeekMask>0111110</dayOfWeekMask>
      <timezoneFlag>1</timezoneFlag>
    </scheduledCommunication>
    <stationary mobilityIndicator="Mobile">
      <trajectory>
      </trajectory>
    </stationary>
  </parameterSet>
</cp:info>
```

6.1.5.1.2 Response

```
HTTP/1.1 201 Created
Date: Mon, 07 Mar 2016 15:17:02 GMT
Location: http://example.com/exampleAPI/communicationpatterns/v1/info/i0003
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<cp:info xmlns:cp="urn:oma:xml:rest:netapi:communicationpatterns:1">
  <serial>89ABCDEF</serial>
  <source>http://www.thirdparty.example.com</source>
  <target>
    <address> tel:+19585550105</address>
    <address>tel:+19585550106</address>
    <address>tel:+19585550107</address>
  </target>
  <validity>
    <startTime>2016-06-23T09:00:00</startTime>
    <expiryTime>2017-06-23T09:00:00</expiryTime>
  </validity>
  <parameterSet>
    <label>Request 3</label>
    <periodicCommunication commIndicator="Periodic">
      <duration>300</duration>
      <interval>7200</interval>
    </periodicCommunication>
    <scheduledCommunication>
      <timeOfDayStart>0</timeOfDayStart>
      <timeOfDayEnd>18000</timeOfDayEnd>
      <dayOfWeekMask>1111100</dayOfWeekMask>
      <timezoneFlag>1</timezoneFlag>
    </scheduledCommunication>
    <scheduledCommunication>
      <timeOfDayStart>72000</timeOfDayStart>
      <timeOfDayEnd>0</timeOfDayEnd>
      <dayOfWeekMask>0111110</dayOfWeekMask>
      <timezoneFlag>1</timezoneFlag>
    </scheduledCommunication>
    <stationary mobilityIndicator="Mobile">
      <trajectory>
      </trajectory>
    </stationary>
  </parameterSet>
<resourceURL>http://example.com/exampleAPI/communicationpatterns/v1/info/i0003</resourceURL>
</cp:info>
```

6.1.6 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 as per sections 6.5.5 and 7.4.1 of [RFC7231].

6.2 Resource: Communication patterns information instance

The resource used is:

http://{serverRoot}/communicationpatterns/{apiVersion}/info/{infoId}

This resources is used for retrieving, updating and deleting communication patterns information related to an infoId.

6.2.1 Request URL variables

The following request URL variables are common for all HTTP methods:

Name	Description
serverRoot	Server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: example.com/exampleAPI
apiVersion	Version of the API client wants to use. The value of this variable is defined in section 5.1
infoId	Identifier of a configuration information

6.2.2 Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common].

For Policy Exception and Service Exception fault codes applicable to Communication Patterns API, see section 7.

6.2.3 GET

This operation is used to retrieve the communication patterns information related to a particular infoId.

6.2.3.1 Example: Retrieve submitted communication patterns information (Informative)

In this example, a query is initiated by an application to retrieve the communication patterns information related to a particular infoId residing in the network server.

6.2.3.1.1 Request

```
GET /exampleAPI/communicationpatterns/v1/info/i0001 HTTP/1.1
Host: example.com
Accept: application/xml
```

6.2.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Thu, 07 Mar 2016 11:00:00 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<cp:info xmlns:cp="urn:oma:xml:rest:netapi:communicationpatterns:1">
 <serial>A00001EF</serial>
  <source>http://www.thirdparty.example.com</source>
  <target>
   <address>tel:+19585550100</address>
   <address>tel:+19585550101</address>
   <address>tel:+19585550102</address>
  </target>
  <validity>
   <startTime>2016-06-23T18:45:00-07:00</startTime>
   <expiryTime>2017-06-23T18:45:00-07:00</expiryTime>
  </validity>
  <parameterSet>
   <label>Request 1</label>
   <periodicCommunication commIndicator="Periodic">
    <duration>300</duration>
    <interval>7200</interval>
   </periodicCommunication>
   <scheduledCommunication>
                 <timeOfDayStart>0</timeOfDayStart>
                 <timeOfDayEnd>18000</timeOfDayEnd>
                 <dayOfWeekMask>1111100</dayOfWeekMask>
                 <timezoneFlag>1</timezoneFlag>
   </scheduledCommunication>
   <scheduledCommunication>
                 <timeOfDayStart>72000</timeOfDayStart>
                 <timeOfDayEnd>0</timeOfDayEnd>
                 <dayOfWeekMask>0111110</dayOfWeekMask>
                 <timezoneFlag>1</timezoneFlag>
   </scheduledCommunication>
   <stationary mobilityIndicator="Mobile">
    <trajectory>
    </trajectory>
   </stationary>
  </parameterSet>
  <resourceURL>http://example.com/exampleAPI/communicationpatterns/v1/info/i0001</resourceURL>
</cp:info>
```

6.2.4 PUT

This operation is used to update the communication patterns information related to a particular infoId.

6.2.4.1 Example: Update submitted communication patterns information (Informative)

In this example, a request is initiated by an application to update the submitted pattern information for an existing infoId.

6.2.4.1.1 Request

```
PUT /exampleAPI/communicationpatterns/v1/info/i0004 HTTP/1.1
Host: example.com
Accept: application/xml
Content-Length: nnnn
MIME-Version: 1.0

<?xml version="1.0" encoding="UTF-8"?>
<cp:info xmlns:cp="urn:oma:xml:rest:netapi:communicationpatterns:1">
  <serial>A00001EF</serial>
    <source>http://www.thirdparty.example.com</source>
    <target>
      <address>tel:+19585550100</address>
      <address>tel:+19585550101</address>
      <address>tel:+19585550102</address>
    </target>
    <validity>
      <startTime>2016-06-23T18:45:00-07:00</startTime>
      <expiryTime>2017-06-23T18:45:00-07:00</expiryTime>
    </validity>
    <parameterSet>
      <label>Request 1</label>
      <periodicCommunication commIndicator="OnDemand">
      </periodicCommunication>
      <scheduledCommunication>
                            <timeOfDayStart>0</timeOfDayStart>
                            <timeOfDayEnd>18000</timeOfDayEnd>
                            <dayOfWeekMask>1111100</dayOfWeekMask>
                            <timezoneFlag>1</timezoneFlag>
      </scheduledCommunication>
      <scheduledCommunication>
                            <timeOfDayStart>72000</timeOfDayStart>
                            <timeOfDayEnd>0</timeOfDayEnd>
                            <dayOfWeekMask>0111110</dayOfWeekMask>
                            <timezoneFlag>1</timezoneFlag>
      </scheduledCommunication>
      <stationary mobilityIndicator="Mobile">
        <trajectory>
        </trajectory>
      </stationary>
    </parameterSet>
</cp:info>
```

6.2.4.1.2 Response

```
HTTP/1.1 200 OK
Date: Thu, 07 Mar 2016 11:00:00 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<cp:info xmlns:cp="urn:oma:xml:rest:netapi:communicationpatterns:1">
  <serial>A00001EF</serial>
    <source>http://www.thirdparty.example.com</source>
    <target>
      <address>tel:+19585550100</address>
      <address>tel:+19585550101</address>
      <address>tel:+19585550102</address>
    </target>
    <validity>
      <startTime>2016-06-23T18:45:00-07:00</startTime>
      <expiryTime>2017-06-23T18:45:00-07:00</expiryTime>
    </validity>
    <parameterSet>
      <label>Request 1</label>
      <periodicCommunication commIndicator="OnDemand">
      </periodicCommunication>
      <scheduledCommunication>
                              <timeOfDayStart>0</timeOfDayStart>
                              <timeOfDayEnd>18000</timeOfDayEnd>
                              <dayOfWeekMask>1111100</dayOfWeekMask>
                              <timezoneFlag>1</timezoneFlag>
      </scheduledCommunication>
      <scheduledCommunication>
                              <timeOfDayStart>72000</timeOfDayStart>
                              <timeOfDayEnd>0</timeOfDayEnd>
                              <dayOfWeekMask>0111110</dayOfWeekMask>
                              <timezoneFlag>1</timezoneFlag>
      </scheduledCommunication>
      <stationary mobilityIndicator="Mobile">
        <trajectory>
        </trajectory>
      </stationary>
    </parameterSet>
    <resourceURL>http://example.com/exampleAPI/communicationpatterns/v1/info/i0004</resourceURL>
</cp:info>
```

6.2.4.2 Example: Update submitted validity time of communication patterns information (Informative)

The operation is used for updating the validity time for a communication pattern information for infoId i0004

6.2.4.2.1 Request

```
PUT / exampleAPI/communicatinpatterns/v1/info/i0004/validity HTTP/1.1
Host: example.com
Accept: application/xml
Content-Length: nnnn
MIME-Version: 1.0

<?xml version="1.0" encoding="UTF-8"?>
<cp:validity xmlns:cp="urn:oma:xml:rest:netapi:communicationPatterns:1">
  <startTime>2016-06-23T19:45:00-07:00</startTime>
  <expiryTime>2017-06-23T19:45:00-07:00</expiryTime>
</cp:validity>
```

6.2.4.2.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Tue, 7 Mar 2016 10:50:00 GMT

<?xml version="1.0" encoding="UTF-8"?>
<cp:validity xmlns:cp="urn:oma:xml:rest:netapi:communicationPatterns:1">
  <startTime>2016-06-23T19:45:00-07:00</startTime>
  <expiryTime>2017-06-23T19:45:00-07:00</expiryTime>
</cp:validity>
```

6.2.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET/POST/DELETE’ field in the response as per sections 6.5.5 and 7.4.1 of [RFC7231].

6.2.6 DELETE

This operation is used to delete the communication patterns information if required.

6.2.6.1 Example: Delete submitted communication patterns information (Informative)

6.2.6.1.1 Request

```
DELETE/exampleAPI/communicatinpatterns/v1/info/i0004 HTTP/1.1
Accept: application/xml
Host: example.com
```

6.2.6.1.2 Response

```
HTTP/1.1 204 No Content
Date: Date: Tue, 7 Mar 2016 10:50:00 GMT
```

6.3 Resource: Configuration status

The resource used is:

://{serverRoot}/communicationpatterns/{apiVersion}/info/{infoId}/configStatus

This resource is used for retrieving status of communication pattern information related to a particular infoId.

6.3.1 Request URL variables

The following request URL variables are common for all HTTP methods:

Name	Description
serverRoot	Server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: example.com/exampleAPI
apiVersion	Version of the API client wants to use. The value of this variable is defined in section 5.1
infoId	Identifier of a configuration information

See section 6 for a statement on the escaping of reserved characters in URL variables.

6.3.2 Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common].

For Policy Exception and Service Exception fault codes applicable to Communication Patterns API, see section 7.

6.3.3 GET

6.3.3.1 Example: Retrieve Communication Patterns configuration status (Informative)

This operation is used for retrieving the configuration status of a communication pattern information of a particular infoId.

6.3.3.1.1 Request

```
GET /exampleAPI/communicatinpatterns/v1/info/i0001/configStatus HTTP/1.1
Accept: application/xml
Host: example.com
```

6.3.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Tue, 7 Mar 2016 10:50:00 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<cp:configStatus xmlns:cp="urn:oma:xml:rest:netapi:communicationpatterns:1">
  <link rel=”InfoReference” href=” http://example.com/exampleAPI/communicationpatterns/v1/info/i0001”/>
  <status>
    <address>tel:+19585550100</address>
    <code>UpdateRequestSent</code>
  </status>
  <status>
    <address>tel:+19585550101</address>
    <code>UpdateRequestQueued</code>
  </status>
  <status>
    <address>tel:+19585550102 </address>
    <code>UpdateImpossible</code>
  </status>
  <resourceURL>http://example.com/exampleAPI/communicationpatterns/v1/info/i0001/configStatus</resourceURL>
</cp:configStatus>
```

6.3.4 PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET’ field in the response as per sections 6.5.5 and 7.4.1 of [RFC7231].

6.3.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET’ field in the response as per sections 6.5.5 and 7.4.1 of [RFC7231].

6.3.6 DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET’ field in the response as per sections 6.5.5 and 7.4.1 of [RFC7231].

7. Fault definitions

7.1 Service Exceptions

For common Service Exceptions refer to [REST_NetAPI_Common]. There are no additional Service Exception codes defined for the RESTful Communication Patterns API.

7.2 Policy Exceptions

For common Policy Exceptions refer to [REST_NetAPI_Common]. There are no additional Policy Exception codes defined for the RESTful Communication Patterns API.

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
n/a	n/a	No prior version

A.2 Candidate Version 1.0 History

Document Identifier	Date	Sections	Description
Draft Versions: REST_NetAPI _CommunicationPatterns-V1_0	18 Jan 2016	All	First Baseline
	19 Feb 2016	1, 4	Incorporated CRs: • OMA-REST_ComPatt-2016-0002R01-CR_Content_for_Scope,_introduction_and_Version_1.0_sections_of_REST_ComPatt_spec
	25 Mar 2016	1, 4, 5.3	Incorporated CRs: • OMA-REST_ComPatt-2016-0003R01-CR_Communication_Patterns_REST_API_resource_tree • OMA-REST_ComPatt-2016-0005R01-CR_Communication_Patterns_REST_API_resource_structure_summary • OMA-REST_ComPatt-2016-0006R01-CR_Communication_Patterns_REST_API_Sequence_diagrams
	07 Apr 2016	2.1, 5.2, 5.3, 6	Incorporated CRs: • OMA-REST_ComPatt-2016-0004R02-CR_Communication_Patterns_REST_API_data_types • OMA-REST_ComPatt-2016-0007R02-CR_XML_examples_for_info_resource_section_6 • OMA-REST_ComPatt-2016-0008-CR_Update_API_Sequence_diagrams
	15 Apr 2016	4.1, 5.2, 6, 7, Appendix B, Appendix C, Appendix D, Appendix E, Appendix F	Incorporated CRs: • OMA-REST_ComPatt-2016-0009-CR_fault_definitions_for_section_7 • OMA-REST_ComPatt-2016-0010-CR_Annex_SCR_tables • OMA-REST_ComPatt-2016-0014-CR_JSON_examples_for_the_annex_section • OMA-REST_ComPatt-2016-0015-CR_description_for_appendix_D_and_E • OMA-REST_ComPatt-2016-0016R01-CR_Content_for_appendix_F_authorisation_aspects • OMA-REST_ComPatt-2016-0017R01-CR_changes_to_section_4.1_version_1.0 • OMA-REST_ComPatt-2016-0018R01-CR_updates_to_configStatus_data_type • OMA-REST_ComPatt-2016-0019-CR_updates_to_XML_examples_in_section_6

Document Identifier	Date	Sections	Description
	05 May 2016	All sections 4.1 5.1 5.2 5.2.2.3 5.3.2 5.3 5.3.3 5.2.2.9 Appendix C Appendix F	Incorporated CRs: • OMA-REST_ComPatt-2016-0020R02-CR_Resolution_of_editorial_review_comments_for_ComPatt_API • OMA-REST_ComPatt-2016-0027R01-CR_resolution_of_CONR_comment_A033 • OMA-REST_ComPatt-2016-0028R01-CR_resolution_of_CONR_comments_A059_and_A061 • MA-REST_ComPatt-2016-0029R01-CR_resolution_of_CONR_comments_A067,_A068_and_A069 • OMA-REST_ComPatt-2016-0030R01-CR_resolution_of_CONR_comment_A046 • OMA-REST_ComPatt-2016-0031R01-CR_Section_4_Introduction_CP_Concepts • OMA-REST_ComPatt-2016-0035-CR_correction_to_section_F.1.2_in_the_Appendix • OMA-REST_ComPatt-2016-0021R01-CR_preset_resource_removal • OMA-REST_ComPatt-2016-0024R02-CR_JSON_CONR_resolutions • OMA-REST_ComPatt-2016-0025R02-CR_XML_examples_CONR_resolutions • OMA-REST_ComPatt-2016-0026R01-CR_Partial_update_CONR_resolutions • OMA-REST_ComPatt-2016-0032R02-CR_CONRR_Data_types_fixes_clarifications • OMA-REST_ComPatt-2016-0034R01-CR_remaining_technical_comments_CONRR
	06 May 2016	5.1	Incorporates CR: OMA-REST_ComPatt-2016-0033R01-CR_resolution_of_remaining_technical_inconsistencies Editorial changes
	09 May 2016	5.1, 5.2.2.1	Corrented errors
Candidate Version: REST_NetAPI _CommunicationPatterns-V1_0	12 May 2016	n/a	Status changed to Candidate by TP TP Ref # OMA-TP-2016-0071R01-INP_RESTNetAPI_Communication_Patterns_V1_0_ERP_for_Candidate_approval

Appendix B. Static Conformance Requirements (Normative)

The notation used in this appendix is specified in [SCRRULES].

B.1 SCR for REST. COMPATT Server

Item	Function	Reference	Requirement
REST-COMPATT-SUPPORT-S-001-M	Support for the RESTful Communication Patterns API	5, 6	
REST-COMPATT-SUPPORT-S-002-M	Support for the XML request & response format	6	
REST-COMPATT-SUPPORT-S-003-M	Support for the JSON request & response format	6, Appendix C	

B.1.1 SCR for REST.COMPATT.INFO Server

Item	Function	Reference	Requirement
REST-COMPATT-INFO-S-001-M	Support for allocation of resource for submitted communication patterns and return to the client the newly created URL and retrieval of list of submitted communication pattern information	6.1	
REST-COMPATT-INFO-S-002-M	Retrieve the list of submitted communication pattern information - GET	6.1.3	
REST-COMPATT-INFO-S-003-M	Support for allocation of new resource for submitted communication patterns information – POST	6.1.5	

B.1.1 SCR for REST.COMPATT.INFOID Server

Item	Function	Reference	Requirement
REST-COMPATT-INFOID-S-001-M	Support for retrieval of submitted communication patterns information, update of submitted communication patterns information and removal of an submitted communication patterns information	6.2	
REST-COMPATT-INFOID-S-002-M	Retrieval of submitted communication patterns information - GET	6.2.3	
REST-COMPATT-INFOID-S-003-M	Update of submitted communication patterns information - PUT	6.2.4	

Item	Function	Reference	Requirement
REST-COMPATT-INFOID-S-004-M	Removal of an submitted communication patterns information – DELETE	6.2.6	

B.1.2 SCR for REST.COMPATT.CONFIGSTATUS Server

Item	Function	Reference	Requirement
REST-COMPATT-CONFIGSTATUS-S-001-M	Support for reading the status of submitted communication patterns information	6.3	
REST-COMPATT-CONFIGSTATUS-S-002-M	Read the status of submitted communication patterns information – GET	6.3.3	

Appendix C. JSON examples (Informative)

JSON (JavaScript Object Notation) is a Light-weight, text-based, language-independent data interchange format. It provides a simple means to represent basic name-value pairs, arrays and objects. JSON is relatively trivial to parse and evaluate using standard JavaScript libraries, and hence is suited for REST invocations from browsers or other processors with JavaScript engines. Further information on JSON can be found at [RFC7159].

The following examples show the request and response for various operations using the JSON data format. The examples follow the XML to JSON serialization rules in [REST_NetAPI_Common]. A JSON response can be obtained by using the content type negotiation mechanism specified in [REST_NetAPI_Common].

For full details on the operations themselves please refer to the section number indicated.

C.1 Retrieving a list of communication patterns information (section 6.1.3.1)

Request:

```
GET /exampleAPI/communicationpatterns/v1/info HTTP/1.1
Host: example.com
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Date: Thu, 07 Mar 2016 11:00:00 GMT
Content-Type: application/json
Content-Length: nnnn


  "cp:infoList": {
    "-xmlns:cp": "urn:oma:xml:rest:netapi:communicationpatterns:1",
    "info": [
      {
        "serial": "A00001EF",
        "source": "http://www.thirdparty.example.com",
        "target": {
          "address": [
            "tel:+19585550100",
            "tel:+19585550101",
            "tel:+19585550102"
          ]
        },
        "validity": {
          "startTime": "2016-06-23T18:45:00-07:00",
          "expiryTime": "2017-06-23T18:45:00-07:00"
        },
        "parameterSet": {
          "label": "Request 1",
          "periodicCommunication": {
            "-commIndicator": "Periodic",
            "duration": "300",
            "interval": "7200"
          },
      "scheduledCommunication": [
       {
        "timeOfDayStart": "0",
        "timeOfDayEnd": "18000",
        "dayOfWeekMask": "1111100",
        "timezoneFlag": "1"
       },
       {
        "timeOfDayStart": "72000",
        "timeOfDayEnd": "0",
        "dayOfWeekMask": "0111110",
        "timezoneFlag": "1"
       }
      ],
      "stationary": {
       "-mobilityIndicator": "Mobile",
       "trajectory": "
     "
      }
     },
     "resourceURL": "http://example.com/exampleAPI/communicationpatterns/v1/info/i0001"
    },
    {
     "serial": "A00001F0",
     "source": "http://www.thirdparty.example.com",
     "target": {
      "address": [
       "tel:+19585550103",
       "tel:+19585550104"
      ]
     },
     "validity": {
      "startTime": "2016-06-23T09:00:00",
      "expiryTime": "2017-06-23T09:00:00"
     },
     "parameterSet": {
      "label": "Request 2",
      "periodicCommunication": {
       "-commIndicator": "Periodic",
       "duration": "300",
       "interval": "3600"
      },
      "scheduledCommunication": [
       {
        "timeOfDayStart": "32400",
        "timeOfDayEnd": "61200",
        "dayOfWeekMask": "1000001",
        "timezoneFlag": "1"
       },
       {
        "timeOfDayStart": "72000",
        "timeOfDayEnd": "0",
        "dayOfWeekMask": "0111110",
        "timezoneFlag": "1"
       }
      ],
          "stationary": {
            "-mobilityIndicator": "Stationary",
            "trajectory": "
        "
          }
        },
        "resourceURL": "http://example.com/exampleAPI/communicationpatterns/v1/info/i0002"
      }
    ],
    "resourceURL": "http://example.com/exampleAPI/communicationpatterns/v1/info"
  }
}
```

C.2 Retrieving a list of communication patterns information, response with a list of resource references (Section 6.1.3.2)

Request:

```
GET /exampleAPI/communicationpatterns/v1/info HTTP/1.1
Host: example.com
Accept: application/xml
```

Response:

```
HTTP/1.1 200 OK
Date: Thu, 07 Mar 2016 11:00:00 GMT
Content-Type: application/json
Content-Length: nnnn

{
  "cp:infoList": {
    "-xmlns:cp": "urn:oma:xml:rest:netapi:communicationpatterns:1",
    "infoReference": [
      { "resourceURL": "http://example.com/exampleAPI/communicationpatterns/v1/info/i0001" },
      { "resourceURL": "http://example.com/exampleAPI/communicationpatterns/v1/info/i0002" }
    ]
  }
}
```

C.3 Submitting new information about predictable communication patterns of Devices (section 6.1.5.1)

Request:

```
POST /exampleAPI/communicationpatterns/v1/info HTTP/1.1
Host: example.com
Accept: application/json
Content-Length: nnnn
MIME-Version: 1.0

{
  "cp:info": {
    "-xmlns:cp": "urn:oma:xml:rest:netapi:communicationpatterns:1",
    "serial": "89ABCDEF",
    "source": "http://www.thirdparty.example.com",
    "target": {
      "address": [
        "tel:+19585550105",
        "tel:+19585550106",
        "tel:+19585550107"
      ]
    },
    "validity": {
      "startTime": "2016-06-23T09:00:00",
      "expiryTime": "2017-06-23T09:00:00"
    },
    "parameterSet": {
      "label": "Request 3",
      "periodicCommunication": {
        "-commIndicator": "Periodic",
        "duration": "300",
        "interval": "7200"
      },
      "scheduledCommunication": [
        {
          "timeOfDayStart": "0",
          "timeOfDayEnd": "18000",
          "dayOfWeekMask": "1111100",
          "timezoneFlag": "1"
        },
        {
          "timeOfDayStart": "72000",
          "timeOfDayEnd": "0",
          "dayOfWeekMask": "0111110",
          "timezoneFlag": "1"
        }
      ],
      "stationary": {
        "-mobilityIndicator": "Mobile",
        "trajectory": "
      "
      }
    }
  }
}
```

Response:

```
HTTP/1.1 201 Created
Date: Mon, 07 Mar 2016 15:17:02 GMT
Location: http://example.com/exampleAPI/communicationpatterns/v1/info/i0003
Content-Type: application/json
Content-Length: nnnn

{
  "cp:info": {
    "-xmlns:cp": "urn:oma:xml:rest:netapi:communicationpatterns:1",
    "serial": "89ABCDEF",
    "source": "http://www.thirdparty.example.com",
    "target": {
      "address": [
        "tel:+19585550105",
        "tel:+19585550106",
        "tel:+19585550107"
      ]
    },
    "validity": {
      "startTime": "2016-06-23T09:00:00",
      "expiryTime": "2017-06-23T09:00:00"
    },
    "parameterSet": {
      "label": "Request 3",
      "periodicCommunication": {
        "-commIndicator": "Periodic",
        "duration": "300",
        "interval": "7200"
      },
      "scheduledCommunication": [
        {
          "timeOfDayStart": "0",
          "timeOfDayEnd": "18000",
          "dayOfWeekMask": "1111100",
          "timezoneFlag": "1"
        },
        {
          "timeOfDayStart": "72000",
          "timeOfDayEnd": "0",
          "dayOfWeekMask": "0111110",
          "timezoneFlag": "1"
        }
      ],
      "stationary": {
        "-mobilityIndicator": "Mobile",
        "trajectory": "
      "
      }
    },
    "resourceURL": "http://example.com/exampleAPI/communicationpatterns/v1/info/i0003"
  }
}
```

C.4 Retrieve submitted communication patterns information (section 6.2.3.1)

Request:

```
GET /exampleAPI/communicationpatterns/v1/info/i0001 HTTP/1.1
Host: example.com
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Date: Thu, 07 Mar 2016 11:00:00 GMT
Content-Type: application/json
Content-Length: nnnn

{
 "cp:info": {
  "-xmlns:cp": "urn:oma:xml:rest:netapi:communicationpatterns:1",
  "serial": "A00001EF",
  "source": "http://www.thirdparty.example.com",
  "target": {
   "address": [
    "tel:+19585550100",
    "tel:+19585550101",
    "tel:+19585550102"
   ]
  },
  "validity": {
   "startTime": "2016-06-23T18:45:00-07:00",
   "expiryTime": "2017-06-23T18:45:00-07:00"
  },
  "parameterSet": {
   "label": "Request 1",
   "periodicCommunication": {
    "-commIndicator": "Periodic",
    "duration": "300",
    "interval": "7200"
   },
   "scheduledCommunication": [
    {
     "timeOfDayStart": "0",
     "timeOfDayEnd": "18000",
     "dayOfWeekMask": "1111100",
     "timezoneFlag": "1"
    },
    {
     "timeOfDayStart": "72000",
     "timeOfDayEnd": "0",
     "dayOfWeekMask": "0111110",
     "timezoneFlag": "1"
    }
   ],
   "stationary": {
    "-mobilityIndicator": "Mobile",
    "trajectory": "
    "
   }
  },
  "resourceURL": "http://example.com/exampleAPI/communicationpatterns/v1/info/i0001"
 }
}
```

C.5 Update submitted communication patterns information (section 6.2.4.1)

Request:

```
PUT /exampleAPI/communicationpatterns/v1/info/i0004 HTTP/1.1
Host: example.com
Accept: application/json
Content-Length: nnnn
MIME-Version: 1.0

{
  "cp:info": {
    "-xmlns:cp": "urn:oma:xml:rest:netapi:communicationpatterns:1",
    "serial": "A00001EF",
    "source": "http://www.thirdparty.example.com",
    "target": {
      "address": [
        "tel:+19585550100",
        "tel:+19585550101",
        "tel:+19585550102"
      ]
    },
    "validity": {
      "startTime": "2016-06-23T18:45:00-07:00",
      "expiryTime": "2017-06-23T18:45:00-07:00"
    },
    "parameterSet": {
      "label": "Request 1",
      "periodicCommunication": { "-commIndicator": "OnDemand" },
      "scheduledCommunication": [
        {
          "timeOfDayStart": "0",
          "timeOfDayEnd": "18000",
          "dayOfWeekMask": "1111100",
          "timezoneFlag": "1"
        },
        {
          "timeOfDayStart": "72000",
          "timeOfDayEnd": "0",
          "dayOfWeekMask": "0111110",
          "timezoneFlag": "1"
        }
      ],
      "stationary": {
        "-mobilityIndicator": "Mobile",
        "trajectory": "
        "
      }
    }
  }
}
```

Response:

```
HTTP/1.1 200 OK
Date: Thu, 07 Mar 2016 11:00:00 GMT
Content-Type: application/json
Content-Length: nnnn

{
  "cp:info": {
    "-xmlns:cp": "urn:oma:xml:rest:netapi:communicationpatterns:1",
    "serial": "A00001EF",
    "source": "http://www.thirdparty.example.com",
    "target": {
      "address": [
        "tel:+19585550100",
        "tel:+19585550101",
        "tel:+19585550102"
      ]
    },
    "validity": {
      "startTime": "2016-06-23T18:45:00-07:00",
      "expiryTime": "2017-06-23T18:45:00-07:00"
    },
    "parameterSet": {
      "label": "Request 1",
      "periodicCommunication": { "-commIndicator": "OnDemand" },
      "scheduledCommunication": [
        {
          "timeOfDayStart": "0",
          "timeOfDayEnd": "18000",
          "dayOfWeekMask": "1111100",
          "timezoneFlag": "1"
        },
        {
          "timeOfDayStart": "72000",
          "timeOfDayEnd": "0",
          "dayOfWeekMask": "0111110",
          "timezoneFlag": "1"
        }
      ],
      "stationary": {
        "-mobilityIndicator": "Mobile",
        "trajectory": "
        "
      }
    },
    "resourceURL": "http://example.com/exampleAPI/communicationpatterns/v1/info/i0004"
  }
}
}
```

C.6 Update submitted validity time of communication patterns information (section 6.2.4.2)

Request

```
PUT / exampleAPI/communicatinpatterns/v1/info/i0004/validity HTTP/1.1
Host: example.com
Accept: application/xml
Content-Length: nnnn
MIME-Version: 1.0

{
  "cp:validity": {
    "-xmlns:cp": "urn:oma:xml:rest:netapi:communicationPatterns:1",
    "startTime": "2016-06-23T19:45:00-07:00",
    "expiryTime": "2017-06-23T19:45:00-07:00"
  }
}
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Tue, 7 Mar 2016 10:50:00 GMT


{
  "cp:validity": {
    "-xmlns:cp": "urn:oma:xml:rest:netapi:communicationPatterns:1",
    "startTime": "2016-06-23T19:45:00-07:00",
    "expiryTime": "2017-06-23T19:45:00-07:00"
  }
}
```

C.7 Delete submitted communication patterns information (section 6.2.6.1)

Request:

```
DELETE/exampleAPI/communicatinpatterns/v1/info/i0004 HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 204 No Content
Date: Date: Tue, 7 Mar 2016 10:50:00 GMT
```

C.8 Retrieve Communication Patterns configuration status (section 6.3.3.1)

Request:

```
GET /exampleAPI/communicatinpatterns/v1/info/i0001/configStatus HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 200 OK
Date: Tue, 7 Mar 2016 10:50:00 GMT
Content-Type: application/json
Content-Length: nnnn

{
  "cp:configStatus": {
    "-xmlns:cp": "urn:oma:xml:rest:netapi:communicationpatterns:1",
    "status": [
      {
        "address": "tel:+19585550100",
        "code": "UpdateRequestSent"
      },
      {
        "address": "tel:+19585550100",
        "code": "UpdateRequestQueued"
      },
      {
        "address": "tel:+19585550100",
        "code": "UpdateImpossible"
      }
    ],
    "link": {
      "-rel": "InfoReference",
      "-href": "http://example.com/exampleAPI/communicationpatterns/v1/info/i0001"
    }
  }
}
```

Appendix D. Operations mapping (Informative)

As this specification does not have a baseline specification, this appendix is empty.

Appendix E. Light-weight Resources (Informative)

The following table lists all Communication Patterns API data structure elements that can be accessed individually as Light-weight Resources.

For each Light-weight Resource, the following information is provided: corresponding root element name, root element type and [ResourceRelPath] string.

Type of Light-weight Resources (and references to data structures)	Element/attribute that can be accessed as Light-weight Resource	Root element name for the Light-weight Resource	Root element type for the Light-weight Resource	[ResourceRelPath] string that needs to be appended to the corresponding Heavy-weight Resource URL
	target	target	TargetDevices	target
	validity	validity	Validity	validity
	parameterSet	parameterSet	ParameterSet	parameterSet

Note: When appending [ResourceRelPath] string to its Heavy-weight Resource URL, all variables within curly brackets “{}” such as: [list of variable names from ResourceRelPath strings] have to be replaced by their real values.

Appendix F. Authorization aspects (Normative)

This appendix specifies how to use the RESTful Communication Patterns API in combination with some authorization frameworks.

F.1 Use with OMA Authorization Framework for Network APIs

The RESTful Communication Patterns API MAY support the authorization framework defined in [Autho4API_10].

A RESTful Communication Patterns API supporting [Autho4API_10]:

- SHALL conform to section D.1 of [REST_NetAPI_Common];
- SHALL conform to this section F.1.

F.1.1 Scope values

F.1.1.1 Definitions

In compliance with [Autho4API_10], an authorization server serving clients requests for getting authorized access to the resources exposed by the RESTful Communication Patterns API:

- SHALL support the scope values defined in the table below;
- MAY support scope values not defined in this specification.

Scope value	Description	For one-time access token
oma_rest_communicationpatterns.all_{apiVersion }	Provide access to all defined operations on the resources in this version of the API. The {apiVersion} part of this identifier SHALL have the same value as the "apiVersion" URL variable which is defined in section 5.1. This scope value is the union of the other scope values listed in next rows of this table.	No
oma_rest_communicationpatterns.info	Provides access to all defined operations for info and {infoId} resources	No
oma_rest_communicationpatterns.configStatus	Provides access to all defined operations on this resource that stores the configuration status information	No

Table 1: Scope values for RESTful Communication Patterns API

F.1.1.2 Downscoping

In the case where the client requests authorization for "oma_rest_communication patterns.all_{apiVersion}" scope, the authorization server and/or resource owner MAY restrict the granted scope to some of the following scope values:

- oma_rest_communicationpatterns.info

F.1.1.3 Mapping with resources and methods

Tables in this section specify how the scope values defined in section F.1.1.1 for the RESTful Communication Patterns API map to the REST resources and methods of this API. In these tables, the root “oma_rest_communicationpatterns.info” of scope values is omitted for readability reasons.

Resource	**URL** **Base URL:** **http://{serverRoot}/CommunicationPatterns/{apiVersion}**	**Section reference**	**HTTP verbs**			
			GET	**PUT**	**POST**	**DELETE**
Communication patterns information	/info	6.1	**all_{apiVersion}** **or** **info**	no	**all_{apiVersion}** **or** **info**	no
Communication patterns information instance	/info/{infoId}	6.2	**all_{apiVersion}** **or** **info**	**all_{apiVersion}** **or** **info**	no	**all_{apiVersion}** **or** **info**
Configuration status	/info/{infoId}/configStatus	6.3	**all_{apiVersion}** **or** **info**	no	no	no
Individual subset of Information about predictable communication patterns of Devices	/info/{infoId}/configStatus/[ResourceRelPath]	6.4	**no**	**all_{apiVersion}** **or** **configStatus**	no	no

Table 2: Required scope values for: info resource, infoId and configStatus

F.1.2 Use of ‘acr:auth’

This section specifies the use of ‘acr:auth’ in place of an end user identifier in a resource URL path.

An ‘acr’ URI of the form ‘acr:auth’, where ‘auth’ is a reserved keyword MAY be used to avoid exposing a real end user identifier in the resource URL path.

Note: ‘acr:auth’ in place of the end user identifier part of a resource URL path is not used in this specification since end user identifier (e.g. {userId}, {address}, etc.) is not part of the resource URL path defined in this specification.