RESTful bindings for Parlay X Web Services – Presence

Candidate Version 1.0 – 11 Jan 2011

Open Mobile Alliance

OMA-TS-ParlayREST_Presence-V1_0-20110111-C

Contents

Figures

Tables

1. Scope

This specification defines a RESTful Presence API using an HTTP protocol binding, based on similar API defined in [3GPP 29.199-14].

2. References

2.1 Normative References

[3GPP 29.199-14] 3GPP Technical Specification, "Open Service Access (OSA); Parlay X Web Services; Part X: Presence (Release 8)", URL:http://www.3gpp.org/

[OMA-DDS-V2.1] "Presence SIMPLE Data Specification", Open Mobile Alliance™, OMA-DDS-Presence_Data_Ext_V2_1, URL:http://www.openmobilealliance.org/

[OMA_REST_TS_Common] "Common definitions and specifications for OMA REST interfaces", Open Mobile Alliance™, OMA-TS_REST_Common-V1_0, URL:http://www.openmobilealliance.org/

[REST_TS_Common] "RESTful bindings for Parlay X Web Services – Common", Open Mobile Alliance™, OMA-TS-ParlayREST_Common-V1_1, URL:http://www.openmobilealliance.org/

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

[RFC2616] "Hypertext Transfer Protocol -- HTTP/1.1", R. Fielding et. al, January 1999, URL:http://www.ietf.org/rfc/rfc2616.txt

[RFC3265] "Session Initiation Protocol (SIP)-Specific Event Notification", A.B. Roach, Jun 2002**,** URL: http://www.ietf.org/rfc/rfc3265.txt

[RFC3857] "A Watcher Information Event Template-Package for the Session Initiation Protocol (SIP)", J. Rosenberg, Aug 2004**,** URL: http://www.ietf.org/rfc/rfc3857.txt

[RFC3863] "Presence Information Data Format (PIDF)", H.Sugano et al., Aug 2004, URL: http://www.ietf.org/rfc/rfc3863.txt

[RFC4119] "A Presence-based GEOPRIV Location Object Format", J.Peterson, December 2005, URL: http://www.ietf.org/rfc/rfc4119.txt

[RFC4234] "Augmented BNF for Syntax Specifications: ABNF". D. Crocker, Ed., P. Overell. October 2005, URL:http://www.ietf.org/rfc/rfc4234.txt

[RFC4627] "The application/json Media Type for JavaScript Object Notation (JSON)", D. Crockford, July 2006, URL: http://www.ietf.org/rfc/rfc4627.txt

[RFC4479] "A data model for Presence", J.Rosenberg, July 2006, URL: http://www.ietf.org/rfc/rfc4479.txt

[RFC4480] "RPID: Rich Presence Extensions to the Presence Information Data Format (PIDF)", H.Schulzrinne, V.Gurbani, P.Kyzivat, J.rosenberg, July 2006, URL: http://www.ietf.org/rfc/rfc4480.txt

[RFC4482] **"**CIPID: Contact Information for the Presence Information Data Format**",** H. Schulzrinne. July 2006, URL: http://www.ietf.org/rfc/rfc4482.txt

[RFC5139] "Revised Civic Location Format for Presence Information Data Format Location Object (PIDF-LO)". M. Thomson, J. Winterbottom, February 2008, URL: http://www.ietf.org/rfc/rfc5139.txt

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

[W3C-URLENC] W3C HTML 2.0 Specification, form-urlencoded Media Type, URL: http://www.w3.org/MarkUp/html-spec/html-spec_8.html#SEC8.2.1

[W3C-XML11] W3C XML 1.1 Specification, URL: http://www.w3.org/TR/xml11/

[XMLSchema1] W3C Recommendation, XML Schema Part 1: Structures Second Edition, URL: http://www.w3.org/TR/xmlschema-1/

[XMLSchema2] W3C Recommendation, XML Schema Part 2: Datatypes Second Edition, URL: http://www.w3.org/TR/xmlschema-2/

2.2 Informative References

[OMADICT] "Dictionary for OMA Specifications", Version 2.8, Open Mobile Alliance™, OMA-ORG-Dictionary-V2_8, URL:http://www.openmobilealliance.org/

[OMA-PRS-RD] "Presence SIMPLE Requirements", Version 2.0, Open Mobile Alliance™, OMA-RD-Presence_SIMPLE-V2_0, URL: http://www.openmobilealliance.org/

[OMA-PRS-AD] "Provisioning Architecture Overview" Version 1.1, Open Mobile Alliance™, OMA-WAP-ProvArch-V1_1, URL: http://www.openmobilealliance.org/

[REST_WP] "White Paper on Guidelines for REST API specifications", Open Mobile Alliance™, OMA-WP-Guidelines_for_REST_API_specifications, 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].

3.3 Abbreviations

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

4. Introduction

The ParlayREST Technical Specification for Presence contains the HTTP protocol binding for the Parlay X Presence Web Services specification, 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, and form-urlencoding).

Generic guidelines for REST API specification development in OMA can be found in [REST_WP].

4.1 Version 1.0

Version 1.0 of Presence ParlayREST API specification supports the following operations:

- Presentity manages presence data with a certain time-to-live
- Presentity manages persistent presence data
- Presentity manages own content
- Presentity retrieves Watchers to its presence data
- Presentity manages authorization rules
- Watcher retrieves presence data about a single Presentity
- Watcher retrieves presence data for Presentities in a Presence list
- Watcher retrieves content from a Presentity
- Presentity retrieves all active subscriptions
- Watcher retrieves all active subscriptions
- Presentity manages subscriptions for Watchers
- Watcher manages presence subscriptions for single Presentities
- Watcher manages presence subscriptions for Presence lists

5. Presence API definition

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

The terms “heavy-weight resource” and “light-weight resource” refer to resources that are used to access and manage a complete document (data structure), respectively parts of a document (part of data structure or an individual element in the data structure.

Common data types, naming conventions, fault definitions and namespaces are defined in [REST_TS_Common] and [OMA_REST_TS_Common]. Presence specific terminology and conventions are defined in [OMA-PRS-RD] and [OMA-PRS-AD].

The remainder of this document is structured as follows:

Section 5 starts with 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.

The remaining subsections in section 5 contain the detailed specification for each of the resources. Each such subsection defines the resource, the request URI variables that are common for all HTTP commands, the possible HTTP response codes, 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 5 use XML as the format for the message body. Form-urlencoded examples are provided in Appendix C, while JSON examples are provided in Appendix D. Appendix B provides the Static Conformance Requirements (SCR). Appendix E lists the Parlay X equivalent method for each supported ParlayREST resource and method combination, where applicable. Finally, Appendix F lists all Presence data structure elements that can be accessed individually as light-weight resources.

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, and MAY support www-form-urlencoded 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 [OMA_REST_TS_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.

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

5.1 Resources Summary

This section summarizes all the resources used by the Presence API.

The figure below visualizes the resource structure defined by this specification. Note that those nodes in the resource tree which have associated HTTP methods defined in this specification are depicted by solid boxes.

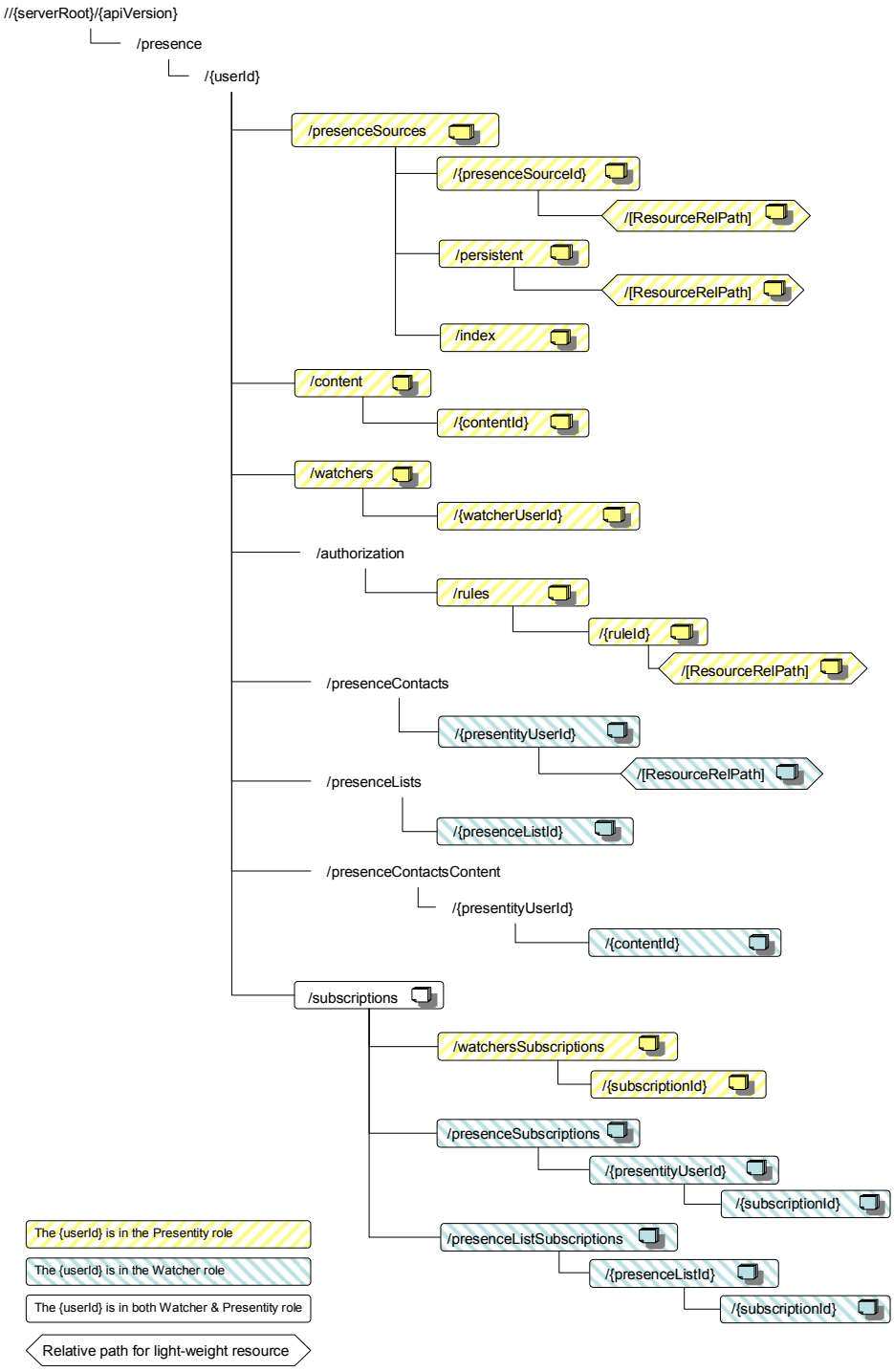

Figure 1 Resource structure defined by this specification

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: Create or update presence data on the server on-behalf of a Presentity

Resource	Base URL: http://example.com/exampleAPI/1/presence	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
Presence sources	/{userId}/presenceSources	PresenceSourceList (Used for GET) PresenceSource (Used for POST) common:ResourceReference (optional alternative for POST response)	Retrieves all presence sources related to a Presentity	No	This operation creates presence source o	No
Individual presence source	/{userId}/presenceSources/{presenceSourceId}	PresenceSource (Used for PUT/GET)	Retrieves the Presentity presence data NOTE: This operation just retrieves the data for this particular presence SourceId and is not the operation a Watcher shall use	Updates presence data for a Presentity	No	Removes the presence data for a Presentity for the specified 'presence SourceId'
Individual presence source attribute	/{userId}/presenceSources/{presenceSourceId}/[ResourceRelPath]	*The data structure corresponds to an element within the PresenceSource structure pointed out by the request-URI.* (Used for PUT/GET)	Retrieves the value of the specified presence attribute	This operation updates a presence attribute	No	This operation removes a presence attribute
Persistent presence source	/{userId}/presenceSources/persistent	PresenceSource	Retrieves the persistent presence data	Creates or updates the persistent presence data	No	Removes the persistent presence data.

Resource	Base URL: http://example.com/exampleAPI/1/presence	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
Individual persistent presence source attribute	/{userId}/presenceSources/persistent/[ResourceRelPath]	*The data structure corresponds to an element within the PresenceSource structure pointed out by the request-URI.* (Used for PUT/GET)	Retrieves the value of the specified presence attribute	This operation creates and updates a persistent presence attribute	No	This operation removes a persistent presence attribute.

Purpose: To allow Presentity to upload and retrieve content (e.g. pictures/avatars/icons)

Resource	Base URL: http://{serverRoot}/{apiVersion}/presence	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
Presentity Content list	/{userId}/content	ContentList	Retrieves all contentIds related to a Presentity	No	No	No
Individual Presentity content	/{userId}/content/{contentId}	*Any MIME content*	Allows Presentity to retrieve its own content (e.g. a picture). NOTE: This operation will retrieve the Presentity's own content and is not the operation a Watcher shall use	Creates/ replaces content on the server	No	Allows a Presentity to remove its own content from the server

Purpose: To allow Presentity to retrieve the list of Watchers interested in the Presentity's presence data

Resource	Base URL: http://{serverRoot}/{apiVersion}/presence	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
Watchers list	/{userId}/watchers	WatcherList	This operation allows the client to retrieve the list of users that are interested in the Presentity's Presence data including the current subscription status	No	No	No
Individual Watcher	/{userId}/watchers/{watcherUserId}	Watcher	This operation allows the Presentity to retrieve the current subscription status and the subscribed attributes for the specified Watcher	No	No	No

Purpose: To allow Presentity to control access to presence information for Watchers, Member lists or Domains.

Resource	Base URL: http://{serverRoot}/{apiVersion}/presence	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
Authorization rules	/{userId}/authorization/rules	RuleList (Used for GET) Rule (Used for POST) common:ResourceReference (optional alternative for POST response)	Retrieves all authorization rules	No	Creates new authorization rule	No
Individual	/{userId}/authorization/ru	Rule (Used for	Retrieves	Updates	No	Removes

authorization rule	les/{ruleId}	PUT/GET)	an authorization rule as specified in the 'ruleId'	an authorization rule as specified in the 'ruleId'		an authorization rule as specified in the 'ruleId'
Individual authorization rule data	/{userId}/authorization/rules/{ruleId}/[ResourceRelPath]	*The data structure corresponds to the element pointed out by the request-URI.* (Used for PUT/GET)	Retrieves the data as specified identified in the URI	Creates or updates data in the rule	No	Removes the data as specified in the URI

Purpose: To allow Watcher to retrieve presence from a single Presentity or a Presence list

Resource	Base URL: http://{serverRoot}/{apiVersion}/presence	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
Presence information by Watcher	/{userId}/presenceContacts/{presentityUserId}	PresenceContact	Retrieves the composite presence from a Presentity, which means that it might include presence data from several presence sources (i.e. after authorization, presence composition and privacy filtering)	No	No	No
Individual presence attribute by Watcher	/{userId}/presenceContacts/{presentityUserId}/[ResourceRelPath]	*The data structure corresponds to an element within the PresenceContact structure pointed out by the request-URI.*	Same as "Presence information by Watcher" but for retrieval of a single presence attribute	No	No	No
Presence information by Watcher for a member list	/{userId}/presenceLists/{presenceListId}	PresenceList	This operation allows a Watcher to retrieve the presence data for all Presentities in a Presence list	No	No	No

Purpose: To allow Watcher to retrieve content from Presentity

Resource	Base URL: http://{serverRoot}/{apiVersion}/presence	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
Content by Watcher	/{userId}/PresenceContactsContent/{presentityUserId}/{contentId}	*Any MIME content*	This operation allows a Watcher to retrieve content (e.g. picture) from another user	No	No	No

Purpose: To allow a user (in both Watcher and Presentity role) to retrieve all its subscriptions

Resource	Base URL: http://{serverRoot}/{apiVersion}/presence	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
All subscriptions	/{userId}/subscriptions	SubscriptionList	Retrieves all active subscriptions	No	No	No

Purpose: To allow Presentity to manage subscriptions for notifications to Watcher Information

Resource	Base URL: http://{serverRoot}/{apiVersion}/presence	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
All Watchers subscriptions	/{userId}/subscriptions/watchersSubscriptions	WatchersSubscriptionList (Used for GET) WatchersSubscription (Used for POST) common:ResourceReference (optional alternative for POST response)	Retrieves all subscriptions related to the Watchers list	No	Creates subscription for the Watchers list	No
Individual Watchers subscription	/{userId}/subscriptions/watchersSubscriptions/{subscriptionId}	WatchersSubscription	Retrieves a subscription to changes in the Watchers list. (A typical usage is to verify if	Updates and/or to extend the duration of the subscription	No	This operation terminates a subscription

			a subscription is still alive)			

Purpose: To allow the server to inform Presentity about updates in Watchers subscription status

Resource	URL: <Specified by the client>	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
Watchers notification	<Specified by the client when the subscription is created or during provisioning process>	WatchersNotification	No	No	This operation is used by the server to inform about updates in Watcher's subscription status	No

Purpose: To allow Watcher to manage subscriptions for presence notifications for a single Presentity

Resource	Base URL: http://{serverRoot}/{apiVersion}/presence	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
All Presence subscriptions	/{userId}/subscriptions/presenceSubscriptions	PresenceSubscriptionList	Retrieves all active presence subscriptions	No	No	No
Presence subscriptions	/{userId}/subscriptions/presenceSubscriptions/{presentityUserId}	PresenceSubscriptionList (Used for GET) PresenceSubscription (Used for POST) common:ResourceReference (optional alternative for POST response)	Retrieves all active presence subscriptions for this particular Presentity	,No	Creates subscription for presence information	No
Individual Presence subscription	/{userId}/subscriptions/presenceSubscriptions/{presentityUserId}/{subscriptionId}	PresenceSubscription (Used for GET/PUT)	Retrieves an active presence subscription. (A typical	Updates and/or to extend the duration of the subscripti	No	This operation terminates a presence subscription

			usage is to verify if a subscription is still alive)	on for presence information		

Purpose: To allow the server to inform Watcher about presence data updates

Resource	URL: <Specified by the client>	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
Presence notification	<Specified by the client when the subscription is created or during provisioning process>	PresenceNotification	No	No	This operation is used by the server to inform about a presence update	No

Purpose: To allow Watcher to manage subscriptions for presence notifications for a Presence list

Resource	Base URL: http://{serverRoot}/{apiVersion}/presence	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
All presence list subscriptions	/{userId}/subscriptions/presenceListSubscriptions	PresenceListSubscriptionCollection	Retrieves all active presence list subscriptions	No	No	No
Presence list subscriptions	/{userId}/subscriptions/presenceListSubscriptions/{presenceListId}	PresenceListSubscriptionCollection (Used for GET) PresenceListSubscription (Used for POST) common:ResourceReference (optional alternative for POST response)	Retrieves all active presence list subscriptions for the specified Presence list	No	Creates a subscription for a Presence list	No
Individual presence list subscription	/{userId}/subscriptions/presenceListSubscriptions/{presenceListId}/{subscriptionId}	PresenceListSubscription (Used for GET/PUT)	Retrieves an individual presence list subscripti	Updates and/or to extend the duration of the presence	No	This operation terminates a presence list subscripti

			on. (A typical usage is to verify if a subscription is still alive)	subscription		on

Purpose: To allow the server to inform Watcher about presence data updates about Presentities in a Presence list

Resource	**URL:** **<Specified by the client>**	**Data Structures**	**HTTP verbs**			
			GET	**PUT**	**POST**	**DELETE**
Presence list notification	<Specified by the client when the subscription is created or during provisioning process>	PresenceListNotification	No	No	This operation is used by the server to inform Watcher about a presence update	No

5.2 Presence ParlayREST API Data Structures

The namespace for the Presence data types is:

urn:oma:xml:rest:presence:1

The 'xsd' namespace is used in the present document to refer to the XML Schema data types defined in XML Schema [XMLSchema1, XMLSchema2]. The 'common' namespace is used in the present document to refer to the data types defined in [REST_TS_Common]. The use of the names 'xsd' and ‘common’ is not semantically significant.

5.2.1 Type: PresenceSourceList

This type describes a list of presence sources.

Element	Type	Optional	Description
presenceSource	PresenceSource [0..unbounded]	Yes	A list of presence sources.
resourceURL	xsd:anyURI	Yes	Self referring URL. SHALL NOT be included in POST requests, MUST be included in responses to any HTTP method that returns an entity body, and in PUT requests.

A root element named presenceSourceList of type PresenceSourceList is allowed in response bodies.

5.2.2 Type: PresenceSource

This type defines a set of parameters for the presence source.

Element	Type	Optional	[ResourceRelPath]	Description
clientCorrelator	xsd:string	Yes	Not applicable	A correlator that the client MAY use to tag this particular resource representation during a request to create a resource on the server. In case the field is present, the server SHALL not alter its value, and SHALL provide it as part of the representation of this resource. In case the field is not present, the server SHALL NOT generate it. This attribute SHALL NOT be present in a document for Persistent Presence. The client SHALL NOT be allowed to update the clientCorrelator in a PUT request.
applicationTag	xsd:string	Yes	Not applicable	A tag that the client MAY use to tag this particular resource on the server. In case the field is present, the server SHALL not alter its value, and SHALL provide it as part of the representation of this resource. In case the field is not present, the server SHALL NOT generate it. This attribute SHALL NOT be present in a document for Persistent Presence.
duration	xsd:int	Yes	duration	Specifies the duration of the publication life time in seconds. When this time has elapsed the subscription will expire unless it has been refreshed. If the parameter is omitted, the publication life time will be specified by the service policy. A too low value (including “0”) will result in an error response. What is too low is defined by service policy. A too high requested value may be reduced by the server according to service policy. This attribute SHALL NOT be present in a document for Persistent Presence.

presence	Presence	Yes	Not applicable	Contains the actual presence attributes. This element SHALL be present in all requests and responses except in the response to GET request with a filter suppressing the element.
resourceURL	xsd:anyURI	Yes	Not applicable	Self referring URL. SHALL NOT be included in POST requests, MUST be included in responses to any HTTP method that returns an entity body, and in PUT requests.

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

Note that the clientCorrelator is used for purposes of error recovery as specified in [REST_TS_Common], and internal client purposes. The server is NOT REQUIRED to use the clientCorrelator value in any form in the creation of the URL of the resource. [REST_TS_Common] provides a recommendation regarding the generation of the value of this field.

Column [ResourceRelPath] includes relative resource paths for light-weight resource URLs that are used to access individual elements in the data structure. A string from this column needs to be appended to the corresponding heavy-weight resource URL in order to create light-weight resource URL for that particular element in the data structure. “Not applicable” means that individual access to that element is not supported. The root element and data type of the resource associated with the [ResourceRelPath] are defined by the Element and Type columns in the row that defines the [ResourceRelPath].

Note that applicationTag is used to enable a particular application instance to pick up (if exists) a previously created resource and continue to operate on it. A typical usage is that a client will perform a GET on the parent resource and in the response receive a list of previously created resources from where the application is able to find its previously created resource. It is up to the client application how to construct the application tag. Please note that a typical usage of the client correlator is not enough for a stateless application to identify a previously created resource since it is uniquely generated every time a new resource is created.

5.2.3 Type: Presence

This type defines a set of presence attributes for a presence source.

Element	Type	Optional	[ResourceRelPath]	Description
person	PersonAttributes	Yes	person	The presence attributes related to person.
service	ServiceAttributes [0..unbounded]	Yes	service/{serviceId}/{version}	The presence attributes related to services. For description of “serviceId” and “version” see 5.2.5. The sub-elements “serviceId” and “version” of the type ServiceAttributes are key properties of service element and SHALL NOT be altered when this element is accessed as a light-weight resource.
device	DeviceAttributes [0..unbounded]	Yes	device/{deviceId}	The presence attributes related to devices. For description of “deviceId” see 5.2.6. The sub-element “deviceId” of the type DeviceAttribute is a key property for device element and SHALL NOT be altered when this element is accessed as a light-weight resource.

Please refer to section 5.2.2 for an explanation of the column [ResourceRelPath].

5.2.4 Type: PersonAttributes

This type defines a set of presence attributes that relate to a person.

Element	Type	Optional	[ResourceRelPath]	Description
activities	Activities	Yes	person/activities	The Presentity's activity (available, busy, lunch, etc.) [RFC4480]
placeType	PlaceType	Yes	person/placeType	At what kind of place the Presentity is (home, office, etc.) [RFC4480]
privacy	Privacy	Yes	person/privacy	The amount of privacy the user wants (public, quiet, etc.) [RFC4480]
sphere	Sphere	Yes	person/sphere	The user's current environment (work, home) [RFC4480]
mood	Mood	Yes	person/mood	The user's mood (angry, confused, happy, etc.) [RFC4480]
placeIs	PlaceIs	Yes	person/placeIs	Describes the properties of the place the user is currently at. [RFC4480]
timeOffset	TimeOffset	Yes	person/timeOffset	Describes the number of minutes of offset from UTC that the user is currently at. [RFC4480]
statusIcon	StatusIcon	Yes	person/statusIcon	Contains a link to an icon of the user. [RFC4480]
class	xsd:token	Yes	person/class	Defines the particular class. [RFC4480]
noteList	NoteList	Yes	person/noteList	Contains taglines of the user. [RFC4479]
location	Location	Yes	person/location	Location of a person. [RFC 5491] and [RFC5139]
overridingWillingness	OverridingWillingness	Yes	person/overridingWillingness	The overriding willingness for a person. [OMA-DDS-V2.1]
linkList	LinkList	Yes	person/linkList	Defines labeled links for a person. [OMA-DDS-V2.1]
card	xsd:anyURI	Yes	person/card	URI to a business card. [RFC4482]
displayName	xsd:string	Yes	person/displayName	A display name of a person. [RFC4482]
homePage	xsd:anyURI	Yes	person/homePage	URI pointing to general information about a person. [RFC4482]
icon	xsd:anyURI	Yes	person/icon	URI pointing to an image/icon of the person. [RFC4482] Note: It is recommended to use the StatusIcon for sharing icons/avatars between users.

map	xsd:anyURI	Yes	person/map	URI pointing to a map related to the person. [RFC4482]
sound	xsd:anyURI	Yes	person/sound	URI pointing to a sound related to the person. [RFC4482]
timestamp	xsd:dateTime	Yes	person/timestamp	Timestamp of the latest update. Mandatory in responses. [RFC3863]
extended	ExtendedList	Yes	person/extended	Attributes for extended presence information

Please refer to section 5.2.2 for an explanation of the column [ResourceRelPath].

5.2.5 Type: ServiceAttributes

This type defines a set of presence attributes that relate to a service.

Element/Attribute	Type	Optional	[ResourceRelPath]	Description
serviceId	xsd:token	No	Not applicable	Identifier of the service. It is a key property of the service. It is defined as an attribute in XML format. NOTE: It is recommended that standards developing organizations making use of the <serviceId> element register it within OMNA Presence <service-description> Registry.
version	xsd:token	No	Not applicable	The version of the specified service. It is a key property of the service. It is defined as an attribute in XML format.
statusIcon	StatusIcon	Yes	service/{serviceId}/{version}/statusIcon	Contains a link to an icon of the user. [RFC4480]
class	xsd:token	Yes	service/{serviceId}/{version}/class	Defines the particular class. [RFC4480]
displayName	xsd:string	Yes	service/{serviceId}/{version}/displayName	A display name of a Service. [RFC4482]
homePage	xsd:anyURI	Yes	service/{serviceId}/{version}/homepage	URI pointing to general information about a Service. [RFC4482]
icon	xsd:anyURI	Yes	service/{serviceId}/{version}/icon	URI pointing to an image/icon of the Service. [RFC4482] Note: It is recommended to use the StatusIcon for sharing icons/avatars between users.
map	xsd:anyURI	Yes	service/{serviceId}/{version}/map	URI pointing to a map related to the Service. [RFC4482]
sound	xsd:anyURI	Yes	service/{serviceId}/{version}/sound	URI pointing to a sound related to the Service. [RFC4482]
linkList	LinkList	Yes	service/{serviceId}/{version}/links	Defines labeled links for a Service. [OMA-DDS-V2.1]
serviceAvailability	OpenOrClosed	Yes	service/{serviceId}/{version}/serviceAvailability	Service specific availability. ([OMA-DDS-V2.1]
serviceWillingness	OpenOrClosed	Yes	service/{serviceId}/{version}/serviceWillingness	Service specific willingness. [OMA-DDS-V2.1]
contact	Contact	Yes	service/{serviceId}/{version}/contact	A contact address for a Service. [RFC3863]
sessionParticipation	OpenOrClosed	Yes	service/{serviceId}/{version}/sessionParticipation	Indicates a participation in a session. [OMA-DDS-V2.1]
registrationState	ActiveOrTerminated	Yes	service/{serviceId}/{version}/registrationState	The registration state for a Service. [OMA-DDS-V2.1]
barringState	ActiveOrTerminated	Yes	service/{serviceId}/{version}/barringState	The barring state for a Service. [OMA-DDS-V2.1]
sessionAnswerMode	AutomaticOrManual	Yes	service/{serviceId}/{version}/sessionAnswerMode	Indicates answer mode for a session. [OMA-DDS-V2.1]

devices	DeviceIdentity List	Yes	service/{serviceId}/{version} /devices	Identify devices which this particular Service is related to. [RFC4479]
timestamp	xsd:dateTime	Yes	service/{serviceId}/{version} /timestamp	Timestamp of the latest update. Mandatory in responses. [RFC3863]
extended	ExtendedList	Yes	service/{serviceId}/{version} /extended	Attributes for extended presence information

Please refer to section 5.2.2 for an explanation of the column [ResourceRelPath].

5.2.6 Type: DeviceAttributes

This type defines a set of presence attributes that relate to a device.

Element/Attribute	Type	Optional	[ResourceRelPath]	Description
deviceId	xsd:anyURI	No	Not applicable	Identifier of the device. [RFC4479] It is a key property of the device. It is defined as an attribute in XML format
class	xsd:token	Yes	device/{deviceId}/class	Defines the particular class. [RFC4480]
location	Location	Yes	device/{deviceId}/location	Location of a device. [RFC 5491] and [RFC5139]
networkAvailability	NetworkAvailability	Yes	device/{deviceId}/networkAvailability	The network availability for a device. [OMA-DDS-V2.1]
timestamp	xsd:dateTime	Yes	device/{deviceId}/timestamp	Timestamp of the latest update. Mandatory in responses. [RFC3863]
extended	ExtendedList	Yes	device/{deviceId}/extended	Attributes for extended presence information

Please refer to section 5.2.2 for an explanation of the column [ResourceRelPath].

5.2.7 Type: ContentList

This type describes a list of content stored on the server.

Element	Type	Optional	Description
content	ContentData [0..unbounded]	Yes	The list of content stored on the server.
resourceURL	xsd:anyURI	Yes	Self referring URL. SHALL NOT be included in POST requests, MUST be included in responses to any HTTP method that returns an entity body, and in PUT requests.

A root element named contentList of type ContentList is allowed in response bodies.

5.2.8 Type: ContentData

This type describes a content stored on the server.

Element	Type	Optional	Description
link	Common:Link	No	Link to the content instance where the actual content is stored.
contentType	xsd:string	Yes	The MIME content type of the stored content (e.g. image/jpg).
eTag	xsd:string	Yes	Version information related to the stored content. Can be used to detect if the content has been updated compared with a previous retrieval of the content.
fSize	xsd:int	Yes	The file size of the content in bytes (e.g. 102400).
resolution	xsd:string	Yes	The resolution of the content (used for instance if the content is an image). The value of the string is of the type “width x height” (e.g. 640x480) where width and height are specified in number of pixels.

5.2.9 Type: WatcherList

This type describes a list of Watchers for presence information.

Element	Type	Optional	Description
watcher	Watcher [0..unbounded]	Yes	Can contain an array of Watchers subscribing for the Presentity
resourceURL	xsd:anyURI	Yes	Self referring URL. SHALL NOT be included in POST requests, MUST be included in responses to any HTTP method that returns an entity body, and in PUT requests.

A root element named watcherList of type WatcherList is allowed in response bodies.

5.2.10 Type: Watcher

This type defines a set of parameters for the Watcher.

Element	Type	Optional	Description
watcherUserId	xsd:anyURI	No	The Watcher subscribing for the data. The value “anonymous@anonymous” is used to indicate that the Watcher has requested that its user identity is not revealed to the Presentity.
displayName	xsd:string	Yes	An optional display name of the Watcher.
resourceStatus	ResourceStatus	No	Describes the state of the Watcher subscription.
subscribedAttribute	xsd:anyURI [0..unbounded]	Yes	Contains a list of relative paths according to the [ResourceRelPath] in sections 5.2.3, 5.2.4, 5.2.5 and 5.2.6.
resourceURL	xsd:anyURI	Yes	Self referring URL. SHALL NOT be included in POST requests, MUST be included in responses to any HTTP method that returns an entity body, and in PUT requests.

A root element named watcher of type Watcher is allowed in response bodies.

5.2.11 Type: RuleList

This type describes a list of authorisation rules.

Element	Type	Optional	Description
rule	Rule [0..unbounded]	Yes	Contains a list of all authorization rules.
resourceURL	xsd:anyURI	Yes	Self referring URL. SHALL NOT be included in POST requests, MUST be included in responses to any HTTP method that returns an entity body, and in PUT requests.

A root element named ruleList of type RuleList is allowed in response bodies.

5.2.12 Type: Rule

This type defines a set of parameters for an authorisation rule.

Element/Attribute	Type	Optional	[ResourceRelPath]	Description
ruleName	xsd:ID	No	*Not applicable*	A name associated with the rule. It is a key property of the rule and when included in the light-weight resource URL then it SHALL NOT be altered. It is defined as an attribute in XML format
watcherUserId	xsd:anyURI [1..unbounded]	Choice	watchers/{watcherUserId}	Contains a list of Watcher identities.
memberListId	xsd:string [1..unbounded]	Choice	memberLists/{memberListId}	Contains a list of member list identities
domainName	xsd:string [1..unbounded]	Choice	domains/{domainName}	Contains a list of domain names.
anonymous	(empty)	Choice	*Not applicable*	Indicates that this rule applies for requests from anonymous.
otherUser	(empty)	Choice	*Not applicable*	Allows the client to specify a default behavior for unknown users.
decision	DefaultDecisionValue	No	*Not applicable*	The authorization decision for the rule.
presenceFilter	xsd:anyURI [0..unbounded]	Yes	*Not applicable*	Contains filter indicating which presence attributes the Watchers are allowed to see. Please refer to the column [ResourceRelPath] in sections 5.2.3, 5.2.4, 5.2.5 and 5.2.6 for possible values of the presenceFilter with the following clarifications: The 'serviceId' MAY be specified using a "*" meaning that the rule applies to all services. The "version" MUST always be specified using "*". The 'deviceId' MAY be specified using a "*" meaning that the rule applies to all devices. An empty or no-existing filter means that the watchers have access to all presence attributes.

resourceURL	xsd:anyURI	Yes	*Not applicable*	Self referring URL. SHALL NOT be included in POST requests, MUST be included in responses to any HTTP method that returns an entity body, and in PUT requests.

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

XSD modeling use a "choice" to select either watcherUserId, memberListId, domainName, anonymous or otherUser.

Please refer to section 5.2.2 for an explanation of the column [ResourceRelPath].

5.2.13 Type: PresenceList

This type describes a list of presence contacts.

Element	Type	Optional	Description
presenceContact	PresenceContact [0..unbounded]	Yes	The presence data structure for each Presentity in the Presence list.
resourceURL	xsd:anyURI	Yes	Self referring URL. SHALL NOT be included in POST requests, MUST be included in responses to any HTTP method that returns an entity body, and in PUT requests.

A root element named presenceList of type PresenceList is allowed in response bodies.

5.2.14 Type: PresenceContact

This type defines a set of parameters for a presence contact.

Element	Type	Optional	Description
presentityUserId	xsd:anyURI	No	Represents the owner of the presence data.
resourceStatus	ResourceStatus	Yes	Indicates the status of the Watcher in relation to the Presentity. This element must only be included when 'PresenceContact' is used within the 'PresenceLists' resource.
presence	Presence	Yes	The actual presence data for the Presentity.
resourceURL	xsd:anyURI	Yes	Self referring URL. SHALL NOT be included in POST requests, MUST be included in responses to any HTTP method that returns an entity body, and in PUT requests.

A root element named presenceContact of type PresenceContact is allowed in response bodies.

5.2.15 Type: SubscriptionList

This type describes a list of subscriptions.

Element	Type	Optional	Description
presenceSubscriptionList	PresenceSubscriptionList	Yes	Contains an array of presence subscriptions for individual users
presenceListSubscriptionCollection	PresenceListSubscriptionCollection	Yes	Contains an array of presence list subscriptions for Presence lists
watcherSubscriptionList	WatcherSubscriptionList	Yes	Contains a list of Watcher subscriptions.
resourceURL	xsd:anyURI	Yes	Self referring URL. SHALL NOT be included in POST requests, MUST be included in responses to any HTTP method that returns an entity body, and in PUT requests.

A root element named subscriptionList of type SubscriptionList is allowed in response bodies.

5.2.16 Type: WatchersSubscriptionList

This type describes a list of subscriptions for Watchers.

Element	Type	Optional	Description
watcherSubscription	WatcherSubscription [0..unbounded]	Yes	May contain an array of Watcher subscriptions
resourceURL	xsd:anyURI	Yes	Self referring URL. SHALL NOT be included in POST requests, MUST be included in responses to any HTTP method that returns an entity body, and in PUT requests.

A root element named watcherSubscriptionList of type WatcherSubscriptionList is allowed in response bodies.

5.2.17 Type: WatcherSubscription

This type defines a set of parameters for a Watcher subscription.

Element	Type	Optional	Description
presentityUserId	xsd:anyURI	Yes	Identifies the Presentity for which the subscription is created towards. Mandatory in responses. The client SHALL NOT be allowed to update the presentityUserId in a PUT request.
callbackReference	common:CallbackReference	No	Client's Notification endpoint and parameters. Contains the callback URL on which notifications will be sent to for the duration of the subscription.
clientCorrelator	xsd:string	Yes	A correlator that the client MAY use to tag this particular resource representation during a request to create a resource on the server. In case the field is present, the server SHALL not alter its value, and SHALL provide it as part of the representation of this resource. In case the field is not present, the server SHALL NOT generate it. The client SHALL NOT be allowed to update the clientCorrelator in a PUT request.
applicationTag	xsd:string	Yes	A tag that the client MAY use to tag this particular resource on the server. In case the field is present, the server SHALL not alter its value, and SHALL provide it as part of the representation of this resource. In case the field is not present, the server SHALL NOT generate it.
duration	xsd:int	Yes	Specifies the duration of the subscription in seconds. When this time has elapsed the subscription will expire unless it has been refreshed. The server shall always include the remaining duration of the subscription in the response. A too high requested value may be reduced by the server according to service policy. If the parameter is omitted, the subscription life time will be specified by the service policy.
resourceStatusFilter	ResourceStatus [0..unbounded]	Yes	Indicates the desired Watcher subscription statuses that the Presentity is interested to get notifications about. If the parameter is omitted or there is an empty filter it means monitoring all states.
frequency	xsd:int	Yes	Maximum frequency of notifications, expressed as minimum time in seconds between notifications.

resourceURL	xsd:anyURI	Yes	Self referring URL. SHALL NOT be included in POST requests, MUST be included in responses to any HTTP method that returns an entity body, and in PUT requests.

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

5.2.18 Type: WatcherNotification

This type defines a set of parameters for the notifications about a Watcher.

Element	Type	Optional	Description
presentityUserId	xsd:anyURI	No	Identifies the Presentity for which the notification is related to. Normally it is the same user who created the subscription.
callbackData	xsd:string	No	CallbackData as passed by the application during the associated subscription
resourceStatus	ResourceStatus	No	Describes the state for the subscription for Watchers.
watcherList	WatcherList	Yes	Contains a list of Watchers including corresponding subscription status. This element is only present if the resourceStatus=Active.
link	common:Link [0..unbounded]	Yes	Link to other resources that are in relationship with the resource.

A root element named watcherNotification of type WatcherNotification is allowed in watcher notification request.

5.2.19 Type: PresenceSubscriptionList

This type describes a list of presence subscriptions.

Element	Type	Optional	Description
presenceSubscription	PresenceSubscription [0..unbounded]	Yes	Can contain an array of presence subscriptions.
resourceURL	xsd:anyURI	Yes	Self referring URL. SHALL NOT be included in POST requests, MUST be included in responses to any HTTP method that returns an entity body, and in PUT requests.

A root element named presenceSubscriptionList of type PresenceSubscriptionList is allowed in response bodies.

5.2.20 Type: PresenceSubscription

This type defines a set of parameters for the subscription for presence information.

Element	Type	Optional	Description
presentityUserId	xsd:anyURI	Yes	Identifies the Presentity for which the subscription is created towards. Mandatory in responses. The client SHALL NOT be allowed to update the presentityUserId in a PUT request.
callbackReference	common:CallbackReference	No	Client's Notification endpoint and parameters. Contains the callback URL on which notifications will be sent to for the duration of the subscription.
clientCorrelator	xsd:string	Yes	A correlator that the client MAY use to tag this particular resource representation during a request to create a resource on the server. In case the field is present, the server SHALL not alter its value, and SHALL provide it as part of the representation of this resource. In case the field is not present, the server SHALL NOT generate it. The client SHALL NOT be allowed to update the clientCorrelator in a PUT request.
applicationTag	xsd:string	Yes	A tag that the client MAY use to tag this particular resource on the server. In case the field is present, the server SHALL not alter its value, and SHALL provide it as part of the representation of this resource. In case the field is not present, the server SHALL NOT generate it.
anonymous	(empty)	Yes	Allows the Watcher to request that its user identity is not revealed to the Presentity.
duration	xsd:int	Yes	Specifies the duration of the subscription in seconds. When this time has elapsed the subscription will expire unless it has been refreshed. The server shall always include the remaining duration of the subscription in the response. A too high requested value may be reduced by the server according to service policy If the parameter is omitted, the subscription life time will be specified by the service policy.

presenceFilter	xsd:anyURI [0..unbounded]	Yes	Allows the Watcher to indicate what presence data it is interested of. The desired attributes are indicated with relative paths according to the [ResourceRelPath] in sections 5.2.3, 5.2.4, 5.2.5 and 5.2.6 with the following clarifications: The 'serviceId', 'version' and 'deviceId' MAY be specified using a "*" meaning that the filter applies to several services and devices respectively. If the parameter is omitted or there is an empty filter it means monitoring of all attributes.
frequency	xsd:int	Yes	Maximum frequency of notifications (expressed as minimum time in seconds between notifications).
resourceURL	xsd:anyURI	Yes	Self referring URL. SHALL NOT be included in POST requests, MUST be included in responses to any HTTP method that returns an entity body, and in PUT requests.

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

5.2.21 Type: PresenceNotification

This type defines a set of parameters for the presence notifications.

Element	Type	Optional	Description
presentityUserId	xsd:anyURI	No	Identifies the Presentity for which the notification is related to.
callbackData	xsd:string	No	CallbackData as passed by the application during the associated subscription
resourceStatus	ResourceStatus	No	Indicates the status of the subscription for the Presentity.
presence	Presence	Yes	The actual presence data for the Presentity. This element is only present if the resourceStatus=Active.
link	common:Link [0..unbounded]	Yes	Link to other resources that are in relationship with the resource.

A root element named presenceNotification of type PresenceNotification is allowed in presence notification request.

5.2.22 Type: PresenceListSubscriptionCollection

This type describes a collection of presence list subscriptions.

Element	Type	Optional	Description
presenceListSubscription	PresenceListSubscription [0..unbounded]	Yes	Can contain an array of presence list subscriptions.
resourceURL	xsd:anyURI	Yes	Self referring URL. SHALL NOT be included in POST requests, MUST be included in responses to any HTTP method that returns an entity body, and in PUT requests.

A root element named presenceListSubscriptionCollection of type PresenceListSubscriptionCollection is allowed in response bodies.

5.2.23 Type: PresenceListSubscription

This type defines a set of parameters for the presence list subscription.

Element	Type	Optional	Description
presenceListId	xsd:anyURI	Yes	Identifies the Presence list for which the subscription is created towards. Mandatory in responses.
callbackReference	common:CallbackReference	No	Client's Notification endpoint and parameters. Contains the callback URL on which notifications will be sent to for the duration of the subscription.
clientCorrelator	xsd:string	Yes	A correlator that the client MAY use to tag this particular resource representation during a request to create a resource on the server. In case the field is present, the server SHALL not alter its value, and SHALL provide it as part of the representation of this resource. In case the field is not present, the server SHALL NOT generate it. The client SHALL NOT be allowed to update the clientCorrelator in a PUT request.
applicationTag	xsd:string	Yes	A tag that the client MAY use to tag this particular resource on the server. In case the field is present, the server SHALL not alter its value, and SHALL provide it as part of the representation of this resource. In case the field is not present, the server SHALL NOT generate it.
anonymous	(empty)	Yes	Allows the Watcher to request that its user identity is not revealed to the Presentity.
duration	xsd:int	Yes	Specifies the duration of the subscription in seconds. When this time has elapsed the subscription will expire unless it has been refreshed. The server shall always include the remaining duration of the subscription in the response. A too high requested value may be reduced by the server according to service policy. If the parameter is omitted, the subscription life time will be specified by the service policy.
presenceFilter	xsd:anyURI [0..unbounded]	Yes	Allows the Watcher to indicate what presence data it is interested of. The desired attributes are indicated with relative paths according to the [ResourceRelPath] in sections 5.2.3, 5.2.4, 5.2.5 and 5.2.6 with the following clarifications: The 'serviceId', 'version' and 'deviceId' MAY be specified using a "*" meaning that the filter applies to several services and devices respectively. If the parameter is omitted or there is an empty filter it means monitoring of all attributes.
frequency	xsd:int	Yes	Maximum frequency of notifications (expressed as minimum time in seconds between notifications).
resourceURL	xsd:anyURI	Yes	Self referring URL. SHALL NOT be included in POST requests, MUST be included in responses to any HTTP method that returns an entity body, and in PUT requests.

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

5.2.24 Type: PresenceListNotification

This type defines a set of parameters for the presence list notifications.

Element	Type	Optional	Description
presenceListId	xsd:anyURI	No	Identifies the Presence list for which the notification is related to.
callbackData	xsd:string	No	CallbackData as passed by the application during the associated subscription
resourceStatus	ResourceStatus	No	Indicates the state of the subscription.
presenceList	PresenceList	Yes	Contains data for each Presentity in the Presence list. This element is only present if the resourceStatus=Active.
link	common:Link [0..unbounded]	Yes	Link to other resources that are in relationship with the resource.

A root element named presenceListNotification of type PresenceListNotification is allowed in presence notification request.

5.2.25 Type: Activities

The type defines a set of parameters for activities. It is inherited from [RFC4480].

Element	Type	Optional	Description
activityValue	ActivityValue [1..unbounded]	No	The value of the attribute as specified in the URI.
note	common:Language String	Yes	A textual description of what the user is currently doing. The language of the text SHOULD be defined by populating the attribute xml:lang of this element.
other	xsd:string	Yes	Only applicable in case activities=ActivitiesOther
from	xsd:dateTime	Yes	Indicates an absolute time from which time the attribute is expected to be valid.
until	xsd:dateTime	Yes	Indicates an absolute time until which time the attribute is expected to be valid.

5.2.26 Type: PlaceType

The type defines a set of parameters for the type of place. It is inherited from [RFC4480].

Element	Type	Optional	Description
placeTypeValue	PlaceTypeValue [1..unbounded]	No	Indicates the type of place the person is currently at.
note	common:Language String	Yes	A comment about the current place the person is located at. The language of the text SHOULD be defined by populating the attribute xml:lang of this element.
other	xsd:string	Yes	A textual description of what type of place the person is located in. Only applicable when the placeTypeList element is set to PlaceOther.
until	xsd:dateTime	Yes	Indicates an absolute time the attribute is expected to be valid.

5.2.27 Type: Privacy

The type defines a set of parameters for privacy. It is inherited from [RFC4480].

Element	Type	Optional	Description
privacyValue	PrivacyValue [1..unbounded]	No	The value of the attribute as specified in the URI.
note	common:LanguageString	Yes	A textual description of the privacy. The language of the text SHOULD be defined by populating the attribute xml:lang of this element.

5.2.28 Type: Sphere

The type defines a set of parameters for the sphere. It is inherited from [RFC4480].

Element	Type	Optional	Description
sphereValue	SphereValue	No	The value of the attribute as specified in the URI.
other	xsd:anyType	Yes	Only applicable in case sphereValue=SphereOther

5.2.29 Type: Mood

The type defines a set of parameters for mood. It is inherited from [RFC4480].

Element	Type	Optional	Description
moodValue	MoodValue [1..unbounded]	No	The value of the attribute as specified in the URI.
note	common:LanguageString	Yes	A textual description of what type of place the person is located in. The language of the text SHOULD be defined by populating the attribute xml:lang of this element.
other	xsd:string	Yes	Only applicable in case activities=MoodOther
until	xsd:dateTime	Yes	Indicates an absolute time the attribute is expected to be valid.

5.2.30 Type: PlaceIs

This type defines the properties of the place the Presentity is currently at, such as the levels of light and noise. This information can be used by the Watcher to determine the type of communication that is likely to be successful.

Element	Type	Optional	Description
placeIsAudio	PlaceIsAudio	Yes	Describes place conditions for audio communication.
placeIsVideo	PlaceIsVideo	Yes	Describes place conditions for video communication.
placeIsText	PlaceIsText	Yes	Describes place conditions for real-time and instant-messaging communication.

5.2.31 Type: TimeOffset

This type defines a set of parameters for the time offset. It describes the number of minutes of offset from UTC that the user is currently at.

Element	Type	Optional	Description
timeOffset	xsd:int	No	Number of minutes of offset from UTC that the user is currently at.
until	xsd:dateTime	Yes	Indicates an absolute time the attribute is expected to be valid.

5.2.32 Type: StatusIcon

This type defines a set of parameters for the status icon. It includes a URI pointing to an image that represents the current status of the user.

Element	Type	Optional	Description
statusIconAddress	xsd:anyURI	No	URL pointing to the content.
contentType	xsd:string	Yes	The content-type related to the content.
eTag	xsd:string	Yes	The Presentity can specify an eTag (i.e. version) of the content allowing the Watcher to detect when the content has been updated.
fSize	xsd:int	Yes	The size of the content.
resolution	xsd:string	Yes	The resolution of the content (used for instance if the content points to an image).
until	xsd:dateTime	Yes	Indicates an absolute time the content is expected to be valid.

5.2.33 Type: NoteList

This type describes a list of notes. The note parameter is inherited from [RFC4479].

Element	Type	Optional	Description
note	common:Language String [1..unbounded]	No	Contains a list of taglines. The language of the text SHOULD be defined by populating the attribute xml:lang of this element.

5.2.34 Type: Location

This defines a set of parameters for the location. It is inherited from [RFC 5491] and [RFC5139].

Element	Type	Optional	Description
circle	CircleData	Choice	Only applicable when value=circle
civicAddress	CivicAddress	Choice	Only applicable when value=civicAddress
retentionExpiry	xsd:dateTime	No	Set the retention expiry value

XSD modelling use a “choice” to select either circle or civicAddress.

5.2.35 Type: CircleData

This defines a set of parameters that describe a circle.

Element	Type	Optional	Description
latitude	xsd:float	No	Latitude of center point
longitude	xsd:float	No	Longitude of center point
radius	xsd:float	Yes	Radius of circle around center point in meters

5.2.36 Type: CivicAddress

This type defines a set of parameters for the civic address. The parameter names are inherited from [RFC5139].

Element	Type	Optional	Description
country	xsd:token	Yes	Two-letter according to [ISO3166a2].
A1	xsd:string	Yes	National subdivisions (state, region, province, prefecture)
A2	xsd:string	Yes	County, parish, gun (JP), district (IN)
A3	xsd:string	Yes	City, township, shi (JP)
A4	xsd:string	Yes	City division, borough, city district, ward, chou (JP)
A5	xsd:string	Yes	Neighborhood, block
A6	xsd:string	Yes	Group of streets below the neighborhood level
PRM	xsd:string	Yes	Road pre-modifier
PRD	xsd:string	Yes	Leading street direction
RD	xsd:string	Yes	Primary road or street
STS	xsd:string	Yes	Street suffix
POD	xsd:string	Yes	Trailing street suffix
POM	xsd:string	Yes	Road post-modifier
RDSEC	xsd:string	Yes	Road section
RDBR	xsd:string	Yes	Road branch
RDSUBBR	xsd:string	Yes	Road sub-branch
HNO	xsd:string	Yes	House number, numeric part only.
HNS	xsd:string	Yes	House number suffix
LMK	xsd:string	Yes	Landmark or vanity address
LOC	xsd:string	Yes	Additional location information
FLR	xsd:string	Yes	Floor
NAM	xsd:string	Yes	Name (residence, business or office occupant)
PC	xsd:string	Yes	Postal code
BLD	xsd:string	Yes	Building (structure)
UNIT	xsd:string	Yes	Unit (apartment, suite)
ROOM	xsd:string	Yes	Room
SEAT	xsd:string	Yes	Seat (desk, cubicle, workstation)
PLC	xsd:string	Yes	Place-type
PCN	xsd:string	Yes	Postal community name
POBOX	xsd:string	Yes	Post office box (P.O. box
ADDCODE	xsd:string	Yes	Additional Code

5.2.37 Type: OverridingWillingness

This type defines a set of parameters for the overriding willingness.

Element/Attribute	Type	Optional	Description
overridingWillingnessValue	OpenOrClosed	No	The overriding willingness for a person
until	xsd:dateTime	Yes	Specifies validity for the attribute. It is defined as an attribute when used in XML format.

5.2.38 Type: LinkList

This type defines a set of parameters for the link list. It enables the client to set one or more links to different type of content and distribute them to its Watchers. It is inherited from [OMA-DDS-V2.1].

Element/Attribute	Type	Optional	Description
link	xsd:anyURI [0..unbounded]	Yes	The address for the linkContains a list of links. A link can contain a URI pointing to any type of resource. NOTE: Not to be confused with the OMA ParlayREST common:Link element When used to address an OMA ParlayREST resource the link element corresponds to the “href” attribute described in [REST_TS_Common].
label	xsd:string	Yes	Label for the link. The Presentity can provide a description of the link. It is defined as an attribute when used in XML format.
priority	xsd:decimal	Yes	Priority for the link. The Presentity can provide a priority used to indicate to the Watcher which link to select first. It is defined as an attribute when used in XML format
contentType	xsd:string	Yes	MIME content type for the link. The Presentity can, if known, specify the content type related to the addressed content allowing the Watcher to detect e.g. if it can render the addressed content. It is defined as an attribute when used in XML format.
rel	xsd:string	Yes	The rel attribute. The Presentity can, if the link is to a ParlayREST resource, specify the rel attribute as described in [REST_TS_Common] section allowing the Watcher to detect e.g. if it can handle the addressed ParlayREST resource. It is defined as an attribute when used in XML format.
eTag	xsd:string	Yes	The ETag value for the addressed content. The Presentity can specify an eTag (i.e. version) of the addressed content allowing the Watcher to detect that the content has been updated in case it is e.g. caching the content. It is defined as an attribute when used in XML format.
fSize	xsd:int	Yes	The file size of the addressed content. The Presentity can specify the size of the addressed content allowing the Watcher to detect e.g. how much bandwidth an upload of the addressed content requires. It is defined as an attribute when used in XML format.
resolution	xsd:string	Yes	The resolution of the addressed content. The Presentity can specify the resolution of the addressed content (used for instance if the link points to an image). It is defined as an attribute when used in XML format.

5.2.39 Type: Contact

This type defines a set of parameters for the contact. It enables the client to set a contact address for the service.

Element/Attribute	Type	Optional	Description
contactAddress	xsd:anyURI	No	A contact address for the service.
priority	xsd:decimal	Yes	Decimal number between 0 and 1 inclusive with at most 3 digits after the decimal point. Higher values indicate higher priority. It is defined as an attribute when used in XML format.

5.2.40 Type: DeviceIdentityList

This type describes a list of device identities. It enables the client to specify a number of device identities related to the particular service.

Element	Type	Optional	Description
deviceId	xsd:anyURI [1..unbounded]	No	A list of device identities related to the service.

5.2.41 Type: NetworkAvailability

This type describes a list of network availabilities. It enables the client to set the network availability for a device.

Element	Type	Optional	Description
network	Network [0…unbounded]	Yes	Represents the availability for a particular network.

5.2.42 Type: Network

This type defines a set of parameters that describe the network and its availability.

Element/Attribute	Type	Optional	Description
connectionStatus	ActiveOrTerminated	No	Indicates the current status of the connection for the corresponding network.
networkMode	HomeOrVisited	Yes	Indicates the current mode of the client connection.
id	xsd:token	No	The identity of the network (e.g. IMS, GSM, GPRS, 802.11x etc). It is defined as an attribute in XML format.

5.2.43 Type: ExtendedList

This type describes a list of extended presence attributes.

Element	Type	Optional	Description
attribute	AttributeValue [1..unbounded]	No	Name of the extended attribute

5.2.44 Type: AttributeValue

This type defines a set of parameters for the presence attribute value. It enables the client to define a name and value for the extended presence attribute.

Element	Type	Optional	Description
name	xsd:string	No	Name of the extended attribute
value	xsd:any	Yes	Value of the extended attribute

5.2.45 Enumeration: ActivityValue

This enumeration defines possible values to describe the type of user activity. It is inherited from [RFC4480].

Enumeration	Description
Appointment	The user has an appointment.
Available	The user is available for communication.
Busy	The user is busy and is only available for urgent matters.
OnThePhone	The user is on the phone.
Steering	The user is driving a car / train / airplane, etc.
Meeting	The user is in a meeting.
Away	No idea what the user is doing, but he is away.
Meal	The user is eating.
Breakfast	The user is having breakfast.
Lunch	The user is having lunch.
Dinner	The user is having dinner.
PermanentAbsence	The user is away and will not return for an extended period.
Vacation	The user is on vacation.
Holiday	A scheduled national or local holiday.
Performance	The user is in a theatre / concert.
InTransit	The user is in the transit area of an (air) port.
Travel	The user is traveling.
Sleeping	The user is sleeping.
LookingForWork	The user is looking for (paid) work.
Playing	The user is occupying him- or her in amusement, sport, or other recreation.
Presentation	The user is giving a presentation, lecture, or participating in a formal round-table discussion.
Shopping	The user is visiting stores in search of goods or Services.
Spectator	The user is observing an event, such as a sports event.
TV	The user is watching television.
Working	The user is engaged in, typically paid, labor, as part of a profession or job.
Worship	The user is participating in religious rites.
ActivitiesUnknown	The activity of the user is unknown.
ActivitiesOther	The user is doing something not in this list.

5.2.46 Enumeration: PlaceTypeValue

This enumeration defines possible values for the type of a place the user is currently at. It is inherited from [RFC4480].

Enumeration	Description
Arena	The user is at an enclosed area used for sports events.
Home	The user is at home.
Office	The user is in an office.
PublicTransport	The user is on public transport.
Street	Walking on the street.
Outdoors	Generally outdoors.
PublicPlace	The user is in a public place.
Hotel	The user is in a hotel.
Theatre	The user is in a theatre or concert.
Restaurant	The user is in a restaurant, coffee shop or, other public dining establishment.
School	The user is at school.
Industrial	The user is in an industrial building.
Quiet	The user is in a quiet area.
Noisy	The user is in a noisy area.
Aircraft	The user is on an aircraft.
Watercraft	The user is on a vessel for travel on water such as a boat or ship.
Automobile	The user is in a car.
Bus	The user is in a bus.
BusStation	The user is in a bus- station.
TrainStation	The user is in a train-station.
ShoppingArea	The user is in a shopping mall or shopping area.
Airport	The user is in an airport.
Train	The user is in a train.
Bank	The user is in a bank.
Bar	The user is in a bar.
Bicycle	The user is on a bicycle.
Café	The user is in a café; usually a small and informal establishment that serves various refreshments (such as coffee); coffee shop.
Classroom	The user is in an academic classroom or lecture hall.
Club	The user is in a dance club, nightclub, or discotheque.
Construction	The user is at a construction site.
ConventionCenter	The user is in a convention center or exhibition hall.
Government	The user is in a government building, such as those used by the legislative, executive, or judicial branches of governments, including court houses, police stations, and military installations.
Hospital	The user is in a hospital, hospice, medical clinic, mental institution, or doctor's office.
Library	The user is in a library.
Motorcycle	The user is on a motorcycle.
Outdoors	The user outside a building, in or into the open air, such as a park or city streets.
Parking	The user is in a parking lot or parking garage.
PlaceOfWorship	The user is at a religious site where congregations gather for religious observances, such as a church, chapel, meetinghouse, mosque, shrine, synagogue, or temple.
Prison	The user is in a prison, penitentiary, jail or a brig.
Residence	The user is in a private or residential setting.
Stadium	The user is in a stadium.
Store	The user is in a shop or store.
Truck	The user is in a truck.

Underway	The user is in a land-, water-, or aircraft that is underway (in motion).
Warehouse	The user is in a warehouse.
Water	The user is in, on, or above bodies of water, such as an ocean, lake, river, canal, or other waterway.
PlaceOther	The user is in a kind of place not listed here.

5.2.47 Enumeration: PrivacyValue

This enumeration defines possible values for privacy. It is inherited from [RFC4480].

Enumeration	Description
PrivacyPublic	The user is surrounded by other people and cannot discuss openly.
PrivacyPrivate	The user is alone and able to talk openly.
PrivacyQuiet	The user is in a quiet environment and cannot talk at all.
PrivacyOther	None of the other values applies.
PrivacyAudio	Inappropriate individuals are not likely to overhear audio communications.
PrivacyText	Inappropriate individuals are not likely to see text communications.
PrivacyVideo	Inappropriate individuals are not likely to see video communications.

5.2.48 Enumeration: SphereValue

This enumeration describes possible values for sphere. It is inherited from [RFC4480].

Enumeration	Description
SphereWork	The user is acting within his work sphere, i.e. as a member of his company
SphereHome	The user is acting within his home sphere, i.e. as a private person.
SphereUnknown	The current sphere is unknown.
SphereOther	The user is acting neither within his work nor within his home sphere.

5.2.49 Enumeration: MoodValue

This enumeration describes possible values for mood. It is inherited from [RFC4480].

Enumeration	Description
Afraid	The user is afraid.
Amazed	The user is amazed.
Angry	The user is angry.
Annoyed	The user is annoyed.
Anxious	The user is anxious.
Ashamed	The user is ashamed.
Bored	The user is bored.
Brave	The user is brave.
Calm	The user is calm.
Cold	The user is cold.
Confused	The user is confused.
Contented	The user is contented.
Cranky	The user is cranky.
Curious	The user is curious.
Depressed	The user is depressed.
Disappointed	The user is disappointed.
Disgusted	The user is disgusted.
Distracted	The user is distracted.
Embarrassed	The user is embarrassed.
Excited	The user is excited.
Flirtatious	The user is flirtatious.

Frustrated	The user is frustrated.
Grumpy	The user is grumpy.
Guilty	The user is guilty.
Happy	The user is happy.
Hot	The user is hot.
Humbled	The user is humbled.
Humiliated	The user is humiliated.
Hungry	The user is hungry.
Hurt	The user is hurt.
Impressed	The user is impressed.
InAwe	The user is in awe.
InLove	The user is in love.
Indignant	The user is indignant.
Interested	The user is interested.
Invincible	The user is invincible.
Jealous	The user is jealous.
Lonely	The user is lonely.
Mean	The user is mean.
MoodUnknown	The user's mood is unknown.
Moody	The user is moody.
Nervous	The user is nervous.
Neutral	The user is neutral.
Offended	The user is offended.
Playful	The user is playful.
Proud	The user is proud.
Relieved	The user is relieved.
Remorseful	The user is remorseful.
Restless	The user is restless.
Sad	The user is sad.
Sarcastic	The user is sarcastic.
Serious	The user is serious.
Shocked	The user is shocked.
Shy	The user is shy.
Sick	The user is sick.
Sleepy	The user is sleepy.
Stressed	The user is stressed.
Surprised	The user is surprised.
Thirsty	The user is thirsty.
Worried	The user is worried.
MoodOther	The user's current mood is not listed here.

5.2.50 Enumeration: PlaceIsAudio

This enumeration defines possible values to describe the place the Presentity is currently at with respect to audio communication. It is inherited from [RFC4480].

Enumeration	Description
Noisy	The user is in a place with a level of background noise that makes audio communications difficult.
Ok	The environmental conditions are suitable.
Quiet	The user is in a place such as a library, restaurant, place of worship, or theatre that discourages noise, conversation, and other distractions.
Unknown	The place attributes are not known.

5.2.51 Enumeration: PlaceIsVideo

This enumeration defines possible values to describe the place the Presentity is currently at with respect to video communication. It is inherited from [RFC4480].

Enumeration	Description
TooBright	The place is too bright for video communication.
Ok	The environmental conditions for video communication are acceptable.
Dark	The place is too dark for video communication.
Unknown	The environmental conditions for video communication are not known.

5.2.52 Enumeration: PlaceIsText

This enumeration defines possible values to describe the place the Presentity is currently at with respect to real-time text and instant messaging. It is inherited from [RFC4480].

Enumeration	Description
Uncomfortable	The place is uncomfortable for typing or other text entry.
Inappropriate	The place is inappropriate for typing or other text entry.
Ok	The environmental conditions are suitable for typing or other text entry.
Unknown	The place attributes for text communication is not known.

5.2.53 Enumeration: OpenOrClosed

This enumeration defines possible values to describe the state of a presence attribute. It is inherited from [OMA-DDS-V2.1].

Element	Description
Open	Indicates an ‘open’ state.
Closed	Indicates a ‘closed’ state.

5.2.54 Enumeration: ActiveOrTerminated

This enumeration defines possible values to describe the state of a presence attribute or network connection. It is inherited from [OMA-DDS-V2.1].

Element	Description
Active	Indicates an ‘active’ state.
Terminated	Indicates a ‘terminated’ state.

5.2.55 Enumeration: AutomaticOrManual

This enumeration defines possible values to describe the mode of a presence attribute. It is inherited from [OMA-DDS-V2.1].

Element	Description
Automatic	Indicates an ‘automatic’ state.
Manual	Indicates a ‘manual’ state.

5.2.56 Enumeration: HomeOrVisited

This enumeration defines possible values to describe client connection mode to the network. It is inherited from [OMA-DDS-V2.1].

Element	Description
Home	Indicates a ‘home’ state.
Visited	Indicates a ‘visited’ state.

5.2.57 Enumeration: ResourceStatus

This enumeration defines possible values to describe the status of the subscription.

Enumeration	Description
Active	Indicates that the subscription is active and authorized. (corresponds to 'active' state in [RFC3265 and RFC3857)
Pending	Indicates that the subscription is awaiting an authorization decision. ('pending', 'waiting' and 'terminated/giveup' state in [RFC3265] and [RFC3857])
TerminatedBlocked	Indicates that the subscription has been terminated. The subscription was blocked. ('terminated/rejected' state in [RFC3265] and [RFC3857])
TerminatedTimeout	Indicates that the subscription has been terminated. The subscription was not refreshed in time before it expired. ('terminated/timeout' state in [RFC3265] and [RFC3857])
TerminatedNoResource	Indicates that the subscription has been terminated. The intended resource does not exist. ('terminated/noresource' state in [RFC3265] and [RFC3857])
TerminatedOther	Indicates that the subscription has been terminated of an unknown reason. ('terminated/probation and terminated/deactivated' state in [RFC3265] and [RFC3857])

5.2.58 Enumeration: DefaultDecisionValue

This enumeration defines possible values for the default authorization decision.

Enumeration	Description
Allow	New Watchers are automatically granted to access presence data about the Presentity.
Block	New Watchers are automatically blocked from seeing any presence data
Politely-block	New Watchers are automatically politely blocked
Confirm	New Watchers have to be manually authorized before being able to get access to the presence data.

5.2.59 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):

- PresenceSourceList
- PresenceSource
- PresenceList
- PresenceContact
- Content
- ContentList
- PresenceSubscriptionList
- PresenceListSubscriptionCollection
- SubscriptionList

- PresenceSubscription
- PresenceNotification
- PresenceListSubscription
- PresenceListNotification
- WatcherList
- Watcher
- WatcherSubscriptionList
- WatcherSubscription
- WatcherNotification
- RuleList
- Rule

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

5.3 Sequence Diagrams

The following sections show three parts of a possible scenario for the usage of the presence API. There are two applications. Both applications have different roles.

- Application 1 acts on behalf of Alice and has the Presentity role.
- Application 2 acts on behalf of Bob and has a Watcher role.

The sequences also try to show the interaction between these different roles.

5.3.1 Application start-up; publish presence, fetch Watcher information, subscribe to Watcher info

This figure below shows a scenario for starting or restarting an application instance of Application 1 on terminal 1 of Alice. Application 1 is a multi-terminal application and can publish different presence status from each of the terminals the application is running on. The sequence shows the following steps.

- Publishing information by application 1 on terminal 1 on behalf of Alice (step 1 - 2)
- Retrieving information about the Watchers of Alice (step 3 - 4)
- Subscribing to Watcher information for Alice, including the corresponding notification (step 5 - 6)

The resources:

- To fetch the list of presenceSources the following resource is used:
  http://{serverRoot}/{apiVersion}/presence/{userId}/presenceSources
- To create a new presenceSource the following resource is used:
  http://{serverRoot}/{apiVersion}/presence/{userId}/presenceSources
- To fetch the current Watchers this resource is used:
  http://{serverRoot}/{apiVersion}/presence/{userId}/watchers
- To fetch the list of subscriptions the following resource is used:
  http://{serverRoot}/{apiVersion}/presence/{userId}/subscriptions/watchersSubscriptions

- To subscribe to changes in the Watcher information the following resource is used:
  http://{serverRoot}/{apiVersion}/presence/{userId}/subscriptions/watchersSubscriptions
- The notification of the Watcher info is done on the callback URL provided by the application.

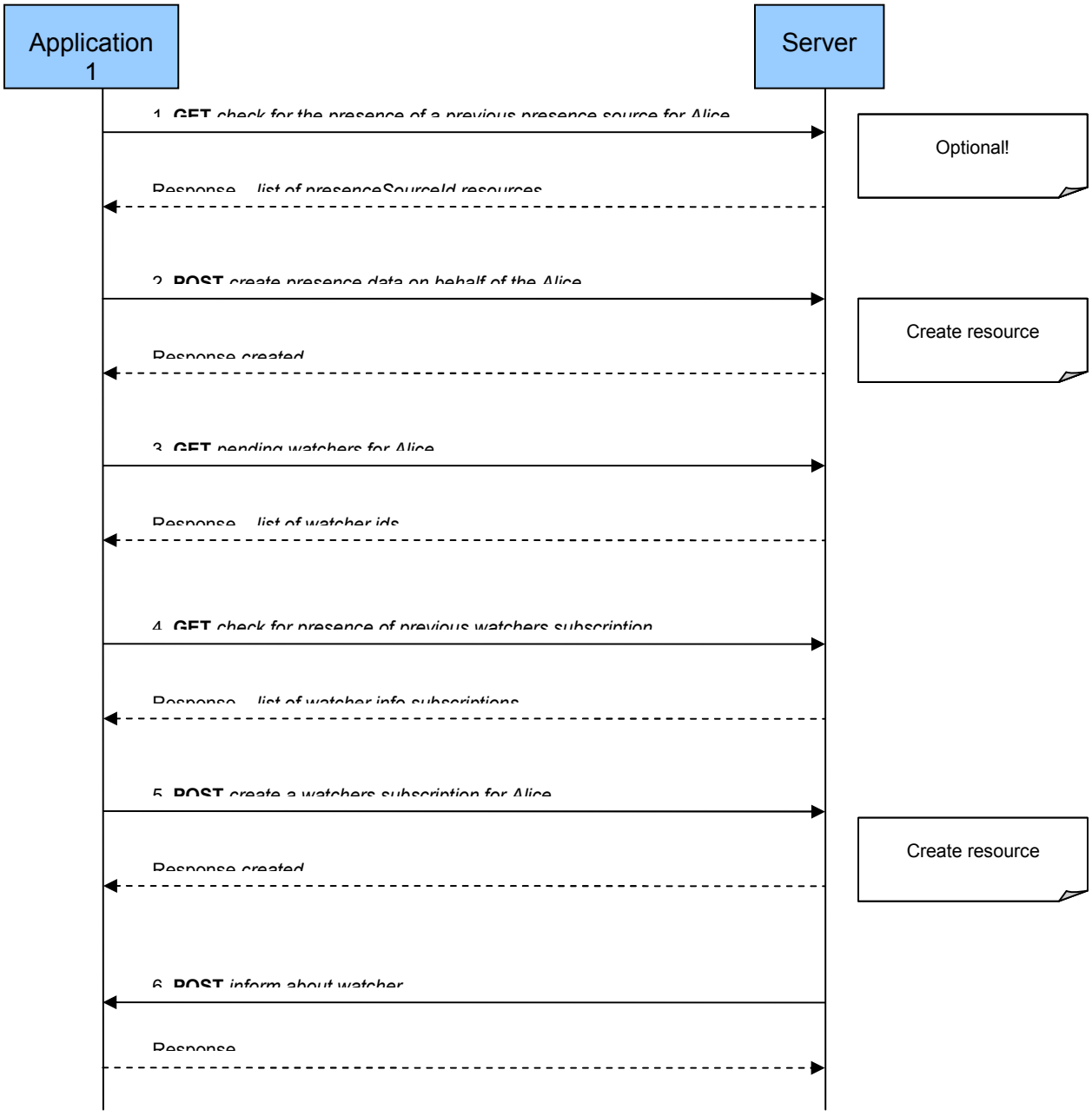

Figure 2 Creation of Presence source, and subscription to Watchers information

Outline of the flows:

The idea is that the application 1 is stateless. i.e., it does not store any data between restarts. So in fact it does not know if the current situation is a start or a restart. The applicationTag is created by the client, and in this case they are created based on

the application id and the terminal id (here: app1_term1), to create a unique identifier per terminal per application. The (optional) applicationTag is used to retrieve resources that were created before the restart and can be reused after the restart.

1. To fetch the list of presenceSources does a GET on the resource:
   http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources

   The response returns a list of presenceSources. Each presenceSource will have a clientCorrelator and an applicationTag. The applications try to find the resource that matches its applicationTag. This way it can find out the resourceURL of that resource.
   In this scenario it is assumed that the resource was not found and the next step is to create a new resource. However, if the resource would have been found, the next step could be to do a GET on the resource, in order to synchronise the client with the server view of the resource (i.e., get the e-tag of the resource and get the current content). After that the client would be in a position to update the resource by using PUT (see later sequences on updating an existing presenceSourceId resource).

2. To create publication data (presenceSourceId) by application 1 on terminal 1 (2) the application does a POST on the following resource to create a new presenceSourceId resource:
   http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSource
   In the POST app1_term1 is provided as the applicationTag. The POST also includes a clientCorrelator that is generated to be unique.

   In the response a 201 result is returned with the location of the resource (we assume here:
   http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/prs123).

3. Application 1 fetches the current Watchers by doing a GET on the following resource
   http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/watchers

   A list of Watchers is returned. The result contains most data about the Watchers, except for some detailed information with is obtained in the following step.

4. Application 1 gets the list of Watcher info subscriptions. This is because in case of a restart it wants to reuse (and probably refresh) the same subscription that was used before the restart. In order to get the list of the subscriptions a GET is performed on:
   http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/subscriptions/watchersSubscriptions

   The response is a list of subscriptions which the application uses to find if there is a subscription matching its specific applicationTag (app1_term1). In this case (after a start) the resource is not found.

5. Application 1 subscribes to changes in the Watcher information by doing a POST on the following resource. The application uses the same applicationTag as used in step 1 and 3 and a unique clientCorrelator.
   http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/subscriptions/watchersSubscriptions

   The response contains a 201 with a location header pointing to the created resource. It is assumed that **http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/subscriptions/watchersSubscriptions/sub001** is returned as the resource.

6. The subscription in step 5 will result in a notification of the application with the current status of the Watcher info. The application provided callback URL is used in the notification.

   This makes step 3 superfluous, but it was included as an alternative way to fetch the same information by polling.

5.3.2 Adding a Watcher; subscribe for presence and updating of presence information.

This is a continuation of the sequence started in the previous section. More specifically the following preconditions apply:

- There is an active subscription for Watcher info by application 1 for the Presentity Alice.

This figure below shows the following scenario:

- Application 2 (a stateful application) subscribes to Alice's presence on behalf of Bob (and corresponding notify) (step 1 - 2)
- Watcher info notification since Bob becomes a pending Watcher (step 3)
- Adding Bob to the allowed list (step 4)
- Presence notification to Bob's application since Bob is now allowed to see the status of Alice (step 5)
- Watcher info notification to Alice's application since the status of the Watcher Bob changed to active (step 6)

The resources:

- To create a subscription for presence notifications for a single entity the following resource is used:
  http://{serverRoot}/{apiVersion}/presence/{userId}/subscriptions/presenceSubscriptions/{presentityUserId}
- The initial notification of the presence information is done on the callback URL provided by the application.
- The notification of the Watchers list is done on the callback URL provided by the application.
- To add a Watcher to the allowed list the following light-weight resource is used:
  http://{serverRoot}/{apiVersion}/presence/{userId}/authorization/rules/{ruleId}/[ResourceRelPath]

  Where [ResourceRelPath] is a light-weight relative resource URL, and in this case it shall be replaced with "watchers/{watcherUserId}"
- The notification of the presence information is done on the callback URL provided by the application.
- The notification of the Watcher info is done on the callback URL provided by the application.

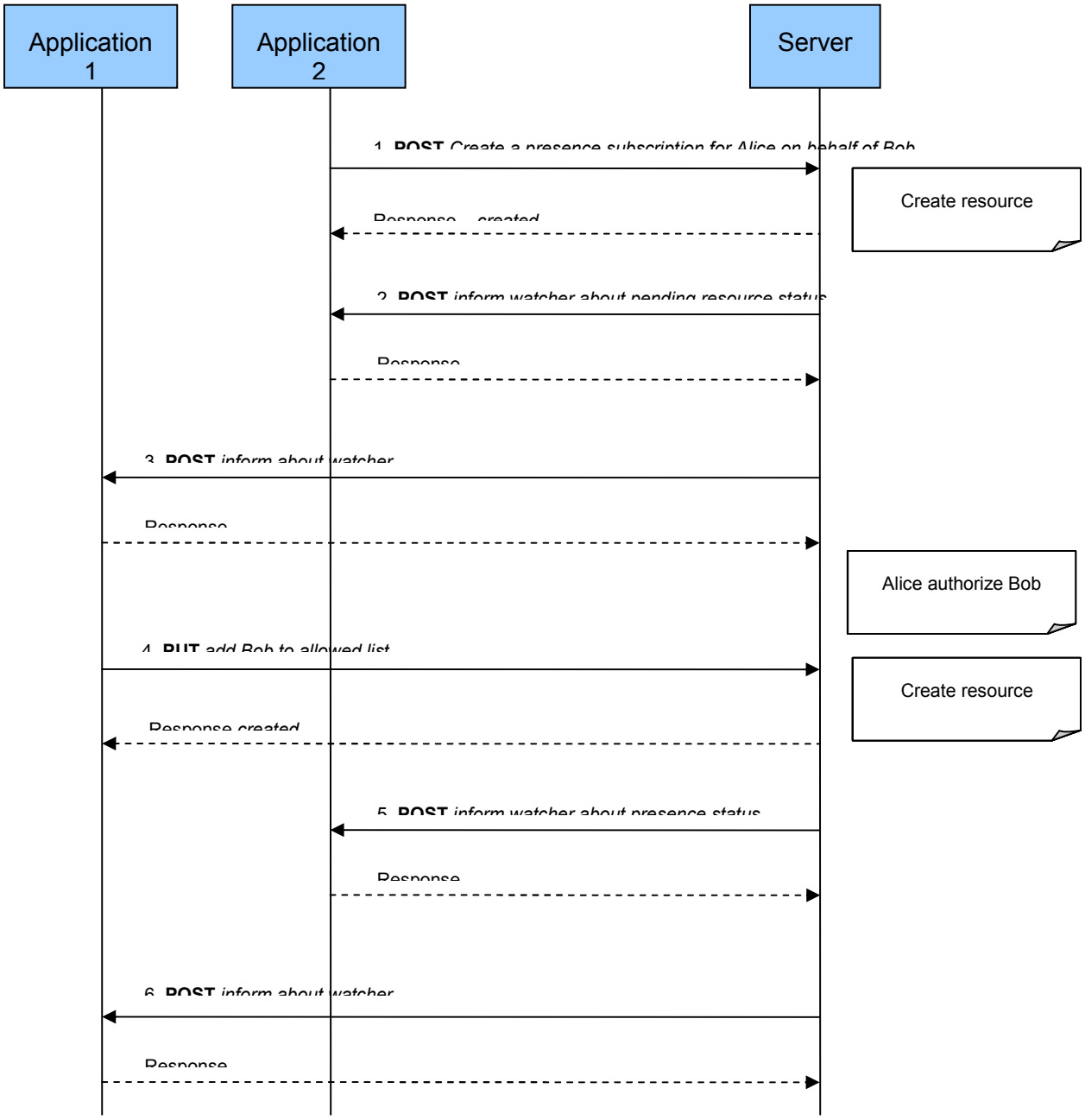

Figure 3 Subscription for Presence data, and Watcher authorization

Outline of the flows:

Application 2 is a stateful application, i.e., it stored information between restarts. Therefore, it will remember the resources that where used in the previous session, and does not have to fetch any resources from the server to find if there is any resource that match its applicationTag.

1. Application 2 creates a subscription to the presence information of Alice. Application 2 acts on behalf of Bob (the Watcher). The subscription is created by doing a POST on a resource with a client generated unique correlator. No applicationTag is included. The resource used for the POST is

http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceSubscriptions/tel%3A%2B1-555-100

As a result a 201 created is returned. The location header pointing to the created resource (here:
http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceSubscriptions/tel%3A%2B1-555-100/sub001

2. The service notifies application 2 about the current status of the subscription. In this case the subscription status is notified as being pending, since Bob is not yet authorized by Alice to view the presence status of Alice.

3. The service notifies application 1 about a new Watcher called Bob, whose status is unauthorized.

4. Application 2 prompts Alice to request authorisation of Bob. Alice allows Bob, so application 2 adds Bob to the allowed list of Alice, meaning that Bob is authorized to view the status of Alice. This is done by performing a PUT on the light-weight resource:
http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/authorization/rules/rule001/watchers/tel%3A%2B1-555-101

   In this case Bob was not yet authorized, so the result is 201 created.

5. The service notifies application 2 about the current status of the subscription. In this case the subscription status is notified as being active, since Bob is now authorized by Alice to view the presence status of Alice.
The notification will also contain the all of the current presence data of Alice that Bob is allowed to see according to the rules.

6. The service notifies application 1 about a new Watcher called Bob, whose status is now changed to active.

5.3.3 Update of presence status

This is a continuation of the sequence started in the previous sections. More specifically the following preconditions apply:

- There is an active subscription for Watcher info by application 1 for the Presentity Alice.
- There is an active subscription for the presence of Presentity Alice by application 2 on behalf of Watcher Bob
- There is an active publication resource for Presentity Alice created by application 1.

This figure below shows the following scenario

- Application 1 uploads a new status-icon for Alice (step 1)
- Application 1 updates the presence data of Alice to with a link to the uploaded status-icon (step 2)
- Application 2 is notified about the changed presence data (step 3)
- Application 2 retrieves the content status-icon (step 4)

The resources:

- To put the content of the status icon the following resource is used:
http://{serverRoot}/{apiVersion}/presence/{userId}/content/{contentId}
- To modify the published presence status the following light-weight resource is used:
http://{serverRoot}/{apiVersion}/presence/{userId}/presenceSources/{presenceSourceId}/[ResourceRelPath]

  Where [ResourceRelPath] is a light-weight relative resource URL, and in this case it shall be replaced with "person/statusIcon"

- The notification of the presence information is done on the callback URL provided by the application.
- To get the content of the status-icon the following resource is used:
  http://{serverRoot}/{apiVersion}/presence/{userId}/presenceContactsContent/{presentityUserId}/{contentId}

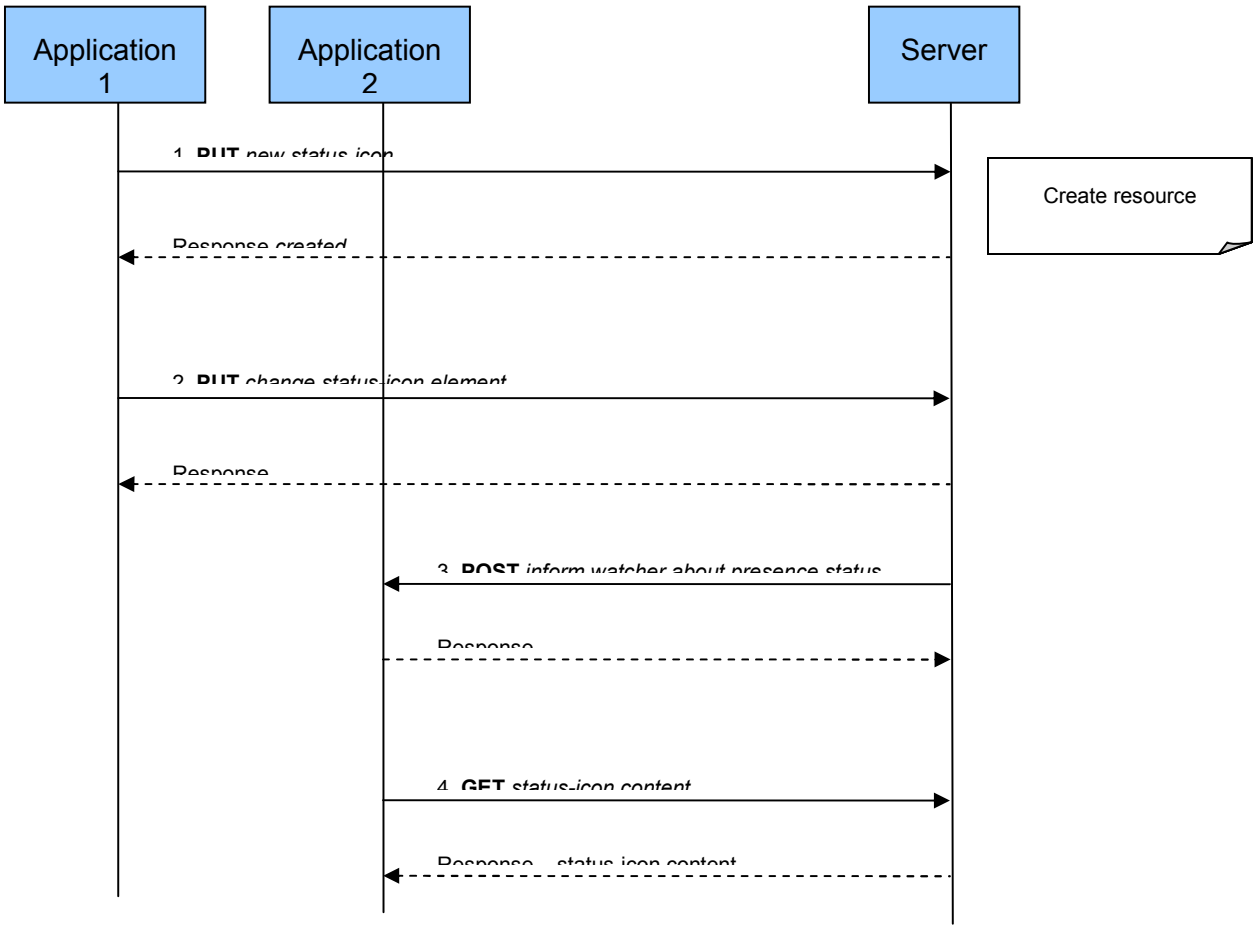

Figure 4 Update of Presence data

Outline of the flows:

1. Application 1 uploads a new status-icon for the Alice. It includes the content of the icon as the body in a PUT on the following resource. This assumes the id of the icon is smiley.
   http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/content/pic001.jpg

   The result depends on whether the content with that ID already exists. In this case it is assumed that it did not yet exist, so a 201 created is returned.

2. Application 1 updates the status of the Alice, by only updating the status-icon part. It does a PUT on the following resource:
   http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/prs123/person/statusIcon

   The result depends on whether the old presence data already contained a status icon.

3. The service notifies Application 2 with the Watcher Bob about the status change of the Presentity Alice. The provided presence information contains the status-icon with a link to the location of the icon

4. Application 2 fetches the content of the status-icon by doing a GET on the following resource:
   http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/presenceContactsContent/tel%3A%2B1-555-100/pic001.jpg

   The response contains the status icon content in the body.

5.3.4 Shutdown; remove resources

This is a continuation of the sequence started in the previous sections. More specifically the following preconditions apply:

- There is an active subscription for Watcher info by application 1 for the Presentity Alice.
- There is an active subscription for the presence of Presentity Alice by application 2 on behalf of Watcher Bob
- There is an active publication resource for Presentity Alice created by application 1.

This figure below shows the following scenario

- All the created subscriptions and the publications are terminated (but not the status-icon content)

The resources:

- To delete the presence subscription the following resource is used:
  http://{serverRoot}/{apiVersion}/presence/{userId}/subscriptions/presenceSubscriptions/{presentityUserId}/{subscriptionId}
- To delete the Watcher info subscription the following resource is used:
  http://{serverRoot}/{apiVersion}/presence/{userId}/subscriptions/watchersSubscriptions/{subscriptionId}
- To delete the publication of presence data the following resource is used:
  http://{serverRoot}/{apiVersion}/presence/{userId}/presenceSources/{presenceSourceId}

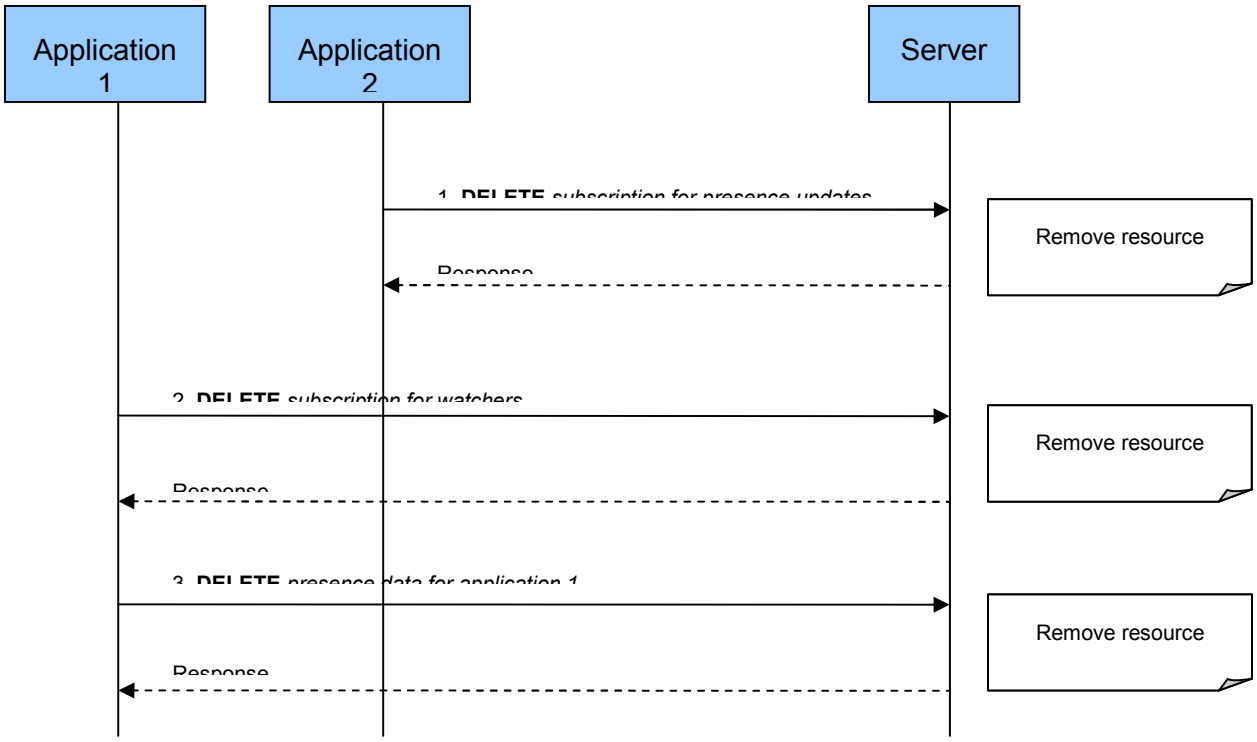

Figure 5 Termination of subscriptions for Watchers, and Presence data

Outline of the flows:

1. Application 2 deletes the subscription resource for presence information by doing a DELETE on the following resource:
   http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceSubscriptions/tel%3A%2B1-555-100/sub001

   Note that a DELETE on a subscription resource will NOT trigger any notifications!

2. Application 1 deletes the Watcher info subscription by doing a DELETE on the following resource:
   http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/subscriptions/watchersSubscriptions/sub001

3. Application 2 deletes the publication of presence data by doing a DELETE on the following resource:
   http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/prs123

5.4 Resource: Presence sources

The resource used is:

http://{serverRoot}/{apiVersion}/presence/{userId}/presenceSources

The userId must be percent-encoded according to [RFC3986].

This resource is used to create a presence source with a life-time. The presence source will expire if it is not refreshed in time. The resource is also used to retrieve all presence sources including the Persistent Presence document.

5.4.1 Request URI variables

The following request URI variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. http://example.com/exampleAPI
apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)
userId	identity of the Presentity that the presence source is created for. Example: tel:+1-555-100

5.4.2 Response Codes

5.4.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.4.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Presence, see Section 6.

5.4.3 GET

This operation is used to retrieve all presence sources for the specified user.

Request URL parameters are:

Name	Type/value	Optional	Description
PresenceSourceFilter	presenceSourceMetaData	Yes	Allows the Presentity to request that only meta data about its Presence Sources are returned in the responds i.e. all presence data is filtered out Example: “?presenceSourceFilter=presenceSourceMetaData”

5.4.3.1 Example: retrieving all presence sources for user (Informative)

5.4.3.1.1 Request

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources HTTP/1.1
Host: example.com:80
Accept: application/xml
```

5.4.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<pr:presenceSourceList xmlns:pr="urn:oma:xml:rest:presence:1">
 <presenceSource>
  <clientCorrelator>123</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <duration>3575</duration>
  <presence>
   <person>
    <mood>
     <moodValue>Happy</moodValue>
    </mood>
   </person>
   <service serviceId="org.openmobilealliance:IM-Session" version="1.0">
    <serviceAvailability>Open</serviceAvailability>
    <devices>
     <deviceId>mac:321</deviceId>
    </devices>
   </service>
   <device deviceId="mac:321">
    <networkAvailability>
     <network id="GPRS">
      <connectionStatus>Active</connectionStatus>
     </network>
    </networkAvailability>
   </device>
  </presence>
  <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/prs123</resourceURL>
 </presenceSource>
 <presenceSource>
  <presence>
   <person>
    <noteList>
     <note xml:lang="en">I am on vacation!</note>
    </noteList>
   </person>
  </presence>
  <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/persistent</resourceURL>
 </presenceSource>
</pr:presenceSourceList>
```

5.4.3.2 Example: retrieving of all presence sources Meta data using a filter (Informative)

5.4.3.2.1 Request

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources?presenceSourceFilter=presenceSourceMetaData HTTP/1.1
Host: example.com:80
Accept: application/xml
```

5.4.3.2.2 Response

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:presenceSourceList xmlns:pr="urn:oma:xml:rest:presence:1">
  <presenceSource>
    <clientCorrelator>123</clientCorrelator>
    <applicationTag>myApp</applicationTag>
    <duration>3575</duration>
    <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/prs123</resourceURL>
  </presenceSource>
  <presenceSource>
    <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/persistent</resourceURL>
  </presenceSource>
</pr:presenceSourceList>
```

5.4.4 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 section 14.7 of [RFC 2616].

5.4.5 POST

This operation is used for creating a presence source with a specified time-to-live. The server might modify the client requested duration to a lower value. If a too low value was requested an error code will be returned.

5.4.5.1 Example 1: creating presence source for user (Informative)

5.4.5.1.1 Request

```
POST /exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources HTTP/1.1
Host: example.com:80
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:presenceSource xmlns:pr="urn:oma:xml:rest:presence:1">
  <clientCorrelator>123</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <duration>7200</duration>
  <presence>
    <person>
      <mood>
        <moodValue>Happy</moodValue>
      </mood>
    </person>
    <service serviceId="org.openmobilealliance:IM-Session" version="1.0">
      <serviceAvailability>Open</serviceAvailability>
      <devices>
        <deviceId>mac:321</deviceId>
      </devices>
```

```
  </service>
  <device deviceId="mac:321">
    <networkAvailability>
      <network id="GPRS">
        <connectionStatus>Active</connectionStatus>
      </network>
    </networkAvailability>
  </device>
 </presence>
</pr:presenceSource>
```

5.4.5.1.2 Response

```
HTTP/1.1 201 Created
Location: http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/prs123
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:presenceSource xmlns:pr="urn:oma:xml:rest:presence:1">
 <clientCorrelator>123</clientCorrelator>
 <applicationTag>myApp</applicationTag>
 <duration>7200</duration>
 <presence>
  <person>
   <mood>
    <moodValue>Happy</moodValue>
   </mood>
  </person>
  <service serviceId="org.openmobilealliance:IM-Session" version="1.0">
   <serviceAvailability>Open</serviceAvailability>
   <devices>
    <deviceId>mac:321</deviceId>
   </devices>
  </service>
  <device deviceId="mac:321">
    <networkAvailability>
      <network id="GPRS">
        <connectionStatus>Active</connectionStatus>
      </network>
    </networkAvailability>
  </device>
 </presence>
 <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/prs123</resourceURL>
</pr:presenceSource>
```

5.4.5.2 Example 2: creating presence source for user fails (Informative)

This example shows a policy exception where the maximum number of publication for Presentity has been reached.

5.4.5.2.1 Request

```
POST /exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources HTTP/1.1
Host: example.com:80
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:presenceSource xmlns:pr="urn:oma:xml:rest:presence:1">
  <clientCorrelator>123</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <duration>7200</duration>
  <presence>
    <person>
      <mood>
        <moodValue>Happy</moodValue>
      </mood>
    </person>
  </presence>
</pr:presenceSource>
```

5.4.5.2.2 Response

```
HTTP/1.1 409 Conflict
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<common:requestError xmlns:common="urn:oma:xml:rest:common:1">
  <link rel="PresenceSourceList"
    href="http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources"/>
  <policyException>
    <messageId>POL0001</messageId>
    <text>A policy error occurred. Error code is %1</text>
    <variables>Max number of presence sources reached</variables>
  </policyException>
</common:requestError>
```

5.4.6 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 section 14.7 of [RFC 2616].

5.5 Resource: Individual presence source

The resource used is:

http://{serverRoot}/{apiVersion}/presence/{userId}/presenceSources/{presenceSourceId}

The userId and presenceSourceId must be percent-encoded according to [RFC3986].

This resource is used for managing of an existing presence source which includes: retrieval of a previously created presence source, update an existing presence source, or to remove a presence source. Only the creator of the presence source SHOULD be allowed to manage it.

5.5.1 Request URI variables

The following request URI variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com/exampleAPI
apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)
userId	identity of the Presentity that the presence source is created for. Example: tel:+1-555-100
presenceSourceId	identifier of the presence source.

5.5.2 Response Codes

5.5.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.5.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Presence, see Section 6.

5.5.3 GET

This operation is used to retrieve a particular presence source for the specified user.

5.5.3.1 Example 1: retrieving presence source (Informative)

5.5.3.1.1 Request

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/prs123 HTTP/1.1
Host: example.com:80
Accept: application/xml
```

5.5.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<pr:presenceSource xmlns:pr="urn:oma:xml:rest:presence:1">
  <clientCorrelator>123</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <duration>5237</duration>
  <presence>
    <person>
      <mood>
```

```
      <moodValue>Happy</moodValue>
    </mood>
  </person>
  <service serviceId="org.openmobilealliance:IM-Session" version="1.0">
   <serviceAvailability>Open</serviceAvailability>
   <devices>
    <deviceId>mac:321</deviceId>
   </devices>
  </service>
  <device deviceId="mac:321">
    <networkAvailability>
     <network id="GPRS">
      <connectionStatus>Active</connectionStatus>
     </network>
    </networkAvailability>
  </device>
 </presence>
 <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/prs123</resourceURL>
</pr:presenceSource>
```

5.5.3.2 Example 2: retrieving presence source which does not exist (Informative)

5.5.3.2.1 Request

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/prs123 HTTP/1.1
Host: example.com:80
Accept: application/xml
```

5.5.3.2.2 Response

```
HTTP/1.1 404 Not Found
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<common:requestError xmlns:common="urn:oma:xml:rest:common:1">
 <link rel="PresenceSourceList"
  href="http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/prs123"/>
 <serviceException>
  <messageId>SVC0001</messageId>
  <text>A service error occurred. Error code is %1</text>
  <variables>Presence source does not exist</variables>
 </serviceException>
</common:requestError>
```

5.5.4 PUT

This operation is used for updating a presence source. The Presentity includes the entire presence document.

5.5.4.1 Example: updating presence source (Informative)

5.5.4.1.1 Request

```
PUT /exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/prs123 HTTP/1.1
Host: example.com:80
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:presenceSource xmlns:pr="urn:oma:xml:rest:presence:1">
 <duration>7200</duration>
 <presence>
  <person>
   <mood>
    <moodValue>Invincible</moodValue>
   </mood>
  </person>
  <service serviceId="org.openmobilealliance:IM-Session" version="1.0">
   <serviceAvailability>Closed</serviceAvailability>
   <devices>
    <deviceId>mac:321</deviceId>
   </devices>
  </service>
  <device deviceId="mac:321">
    <networkAvailability>
     <network id="GPRS">
      <connectionStatus>Active</connectionStatus>
     </network>
    </networkAvailability>
  </device>
 </presence>
</pr:presenceSource>
```

5.5.4.1.2 Response

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:presenceSource xmlns:pr="urn:oma:xml:rest:presence:1">
 <duration>7200</duration>
 <presence>
  <person>
   <mood>
    <moodValue>Invincible</moodValue>
   </mood>
  </person>
  <service serviceId="org.openmobilealliance:IM-Session" version="1.0">
   <serviceAvailability>Closed</serviceAvailability>
   <devices>
    <deviceId>mac:321</deviceId>
   </devices>
```

```
  </service>
  <device deviceId="mac:321">
    <networkAvailability>
      <network id="GPRS">
        <connectionStatus>Active</connectionStatus>
      </network>
    </networkAvailability>
  </device>
 </presence>
 <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/prs123</resourceURL>
</pr:presenceSource>
```

5.5.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: PUT, POST, DELETE' field in the response as per section 14.7 of [RFC 2616].

5.5.6 DELETE

This operation is used for removing a presence source. The Presentity includes the entire presence document.

5.5.6.1 Example: removing presence source (Informative)

5.5.6.1.1 Request

```
DELETE /exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/prs123 HTTP/1.1
Host: example.com:80
```

5.5.6.1.2 Response

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

5.6 Resource: Individual presence source attribute

The resource used is:

http://{serverRoot}/{apiVersion}/presence/{userId}/presenceSources/{presenceSourceId}/[ResourceRelPath]

The userId, presenceSourceId and ResourceRelPath must be percent-encoded according to [RFC3986].

This resource is used to update a particular presence attribute as well as extending the duration of the presence source.

This resource type can be used to access and manage parts of presence attributes. The resource URL consists of heavy-weight resource path, **http://{serverRoot}/{apiVersion}/presence/{userId}/presenceSources/{presenceSourceId}/,** and an extension of the resource URL path, which is relative resource path for a light-weight resource, and it is represented by [**ResourceRelPath].**

5.6.1 Request URI variables

The following request URI variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com/exampleAPI
apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)
userId	identity of the Presentity that the presence source is created for. Example: tel:+1-555-100
presenceSourceId	identifier of the presence source.
[ResourceRelPath]	Relative resource path for a light-weight resource, consisting of a relative path down to an element in the data structure. For more information about the applicable values (strings) for this variable see 5.6.1.1.

5.6.1.1 Light-weight relative resource paths

The following table describes the types of light-weight resources that can be accessed by using this resource, applicable methods, and links to data structures that contain values (strings) for those relative resource paths.

Light-weight resource type	Method supported	Description
Presence attribute groups	GET, PUT, DELETE	Enables access to presence attributes related to Person, Service or Device. See data structure 5.2.3 for possible values for the light-weight relative resource path.
Person attributes	GET, PUT, DELETE	Enables access to a single presence attribute related to a person. See data structure 5.2.4 for possible values for the light-weight relative resource path.
Service attributes	GET, PUT, DELETE	Enables access to a single presence attribute related to a service. See data structure 5.2.5 for possible values for the light-weight relative resource path.
Device attributes	GET, PUT, DELETE	Enables access to a single presence attribute related to a device. See data structure 5.2.6 for possible values for the light-weight relative resource path.
Duration of presence source	GET, PUT	Used to update the duration or retrieve the remaining life time of the presence source. See data structure 5.2.2 for the light-weight relative resource path.

5.6.2 Response Codes

5.6.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.6.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Presence, see Section 6.

5.6.3 GET

This operation is used to retrieve a particular presence attribute in the specified presence source. It may also be used to retrieve the remaining duration of the life time. If the presence source is not refreshed in time it will expire.

5.6.3.1 Example: retrieving individual presence attribute (Informative)

5.6.3.1.1 Request

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/prs123/person/mood HTTP/1.1
Host: example.com:80
Accept: application/xml
```

5.6.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:mood xmlns:pr="urn:oma:xml:rest:presence:1">
  <moodValue>Happy</moodValue>
</pr:mood>
```

5.6.4 PUT

This operation is used to update (or create if it does not exist already) an individual presence attribute in the specified presence source. It may also be used to extend the duration of a presence source.

5.6.4.1 Example: updating individual presence attribute (Informative)

5.6.4.1.1 Request

```
PUT /exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/prs123/person/mood HTTP/1.1
Host: example.com:80
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:mood xmlns:pr="urn:oma:xml:rest:presence:1">
  <moodValue>Excited</moodValue>
</pr:mood>
```

5.6.4.1.2 Response

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:mood xmlns:pr="urn:oma:xml:rest:presence:1">
  <moodValue>Excited</moodValue>
</pr:mood>
```

5.6.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, PUT, DELETE' field in the response as per section 14.7 of [RFC 2616].

5.6.6 DELETE

This operation is used to remove a particular presence attribute from a presence source.

5.6.6.1 Example: removing individual presence attribute (Informative)

5.6.6.1.1 Request

```
DELETE /exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/prs123/person/mood HTTP/1.1
Host: example.com:80
```

5.6.6.1.2 Response

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

5.7 Resource: Persistent presence source

The resource used is:

http://{serverRoot}/{apiVersion}/presence/{userId}/presenceSources/persistent

The userId must be percent-encoded according to [RFC3986].

This resource is used by the Presentity to manage persistent presence data. Persistent presence data is normally used for more static kind of presence data and does not have a time-to-live and hence will never expires. There is only one instance of the persistent presence source in the system.

5.7.1 Request URI variables

The following request URI variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com/exampleAPI
apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)

userId	identity of the Presentity that the persistent presence source is created for. Example: tel:+1-555-100

5.7.2 Response Codes

5.7.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.7.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Presence, see Section 6.

5.7.3 GET

This operation is used to retrieve the persistent presence source for the specified user.

5.7.3.1 Example: retrieving persistent presence data (Informative)

5.7.3.1.1 Request

This example shows also an alternative way to indicate desired content type in response from the server, by using URL query parameter "?resFormat" which is described in [OMA_REST_TS_Common].

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/persistent?resFormat=XML HTTP/1.1
Host: example.com:80
```

5.7.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
ETag:"11"
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<pr:presenceSource xmlns:pr="urn:oma:xml:rest:presence:1">
   <presence>
     <person>
       <noteList>
         <note xml:lang="en">Im on vacation!</note>
       </noteList>
     </person>
   </presence>
   <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/persistent</resourceURL>
</pr:presenceSource>
```

5.7.4 PUT

This operation is used to update (or create if it does not exist already) the persistent presence source for the specified user.

5.7.4.1 Example: updating persistent presence data (Informative)

This example illustrates a scenario where two clients operate on the persistent presence data and are using conditional headers to prevent one client overwriting the data created by another client.

5.7.4.1.1 Request

```
PUT /exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/persistent HTTP/1.1
Host: example.com:80
If-Match: "10"
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:presenceSource xmlns:pr="urn:oma:xml:rest:presence:1">
  <presence>
   <person>
    <statusIcon>
     <statusIconAddress>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/content/pic001.jpg</statusIconAddress>
     <contentType>image/jpg</contentType>
     <eTag>123</eTag>
    </statusIcon>
    <noteList>
     <note xml:lang="en">My picture is updated!</note>
    </noteList>
   </person>
  </presence>
</pr:presenceSource>
```

5.7.4.1.2 Response

```
HTTP/1.1 412  Precondition Failed
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

The request above failed because the other client has modified the resource since the last operation on it, which has led to the server updating the ETag. The client has to retrieve the resource (please refer to the example in 5.7.3) in order to synchronize the ETag again. The response is examined (in order to possibly retain other data), and a new PUT request with the latest ETag value is initiated.

5.7.4.1.3 Request

```
PUT /exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/persistent HTTP/1.1
Host: example.com:80
If-Match: "11"
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:presenceSource xmlns:pr="urn:oma:xml:rest:presence:1">
  <presence>
   <person>
    <statusIcon>
     <statusIconAddress>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/content/pic001.jpg</statusIconAddress>
     <contentType>image/jpg</contentType>
     <eTag>123</eTag>
```

```
      </statusIcon>
      <noteList>
        <note xml:lang="en">My picture is updated!</note>
      </noteList>
    </person>
  </presence>
</pr:presenceSource>
```

5.7.4.1.4 Response

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
ETag: "12"
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:presenceSource xmlns:pr="urn:oma:xml:rest:presence:1">
  <presence>
    <person>
      <statusIcon>
        <statusIconAddress>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/content/pic001.jpg</statusIconAddress>
        <contentType>image/jpg</contentType>
        <eTag>123</eTag>
      </statusIcon>
      <noteList>
        <note xml:lang="en">My picture is updated!</note>
      </noteList>
    </person>
  </presence>
  <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/persistent</resourceURL>
</pr:presenceSource>
```

5.7.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, PUT, DELETE' field in the response as per section 14.7 of [RFC 2616].

5.7.6 DELETE

This operation is used to remove the persistent presence source.

5.7.6.1 Example: removing persistent presence data (Informative)

5.7.6.1.1 Request

```
DELETE /exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/persistent HTTP/1.1
Host: example.com:80
```

5.7.6.1.2 Response

HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT

5.8 Resource: Individual persistent presence source attribute

The resource used is:

http://{serverRoot}/{apiVersion}/presence/{userId}/presenceSourcespersistent/[ResourceRelPath]

The userId and ResourceRelPath must be percent-encoded according to [RFC3986].

This resource is used to manage individual persistent presence attributes. Persistent presence data is normally used for more static kind of data and does not have a time-to-live and hence will never expires.

This resource type can be used to access and manage parts of presence attributes. The resource URL consists of heavy-weight resource path, **http://{serverRoot}/{apiVersion}/presence/{userId}/presenceSources/persistent/,** and an extension of the resource URL path, which is relative resource path for a light-weight resource, and it is represented by [**ResourceRelPath].**

5.8.1 Request URI variables

The following request URI variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com/exampleAPI
apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)
userId	identity of the Presentity that the persistent presence attributes is managed for. Example: tel:+1-555-100
[ResourceRelPath]	Relative resource path for a light-weight resource, consisting of a relative path down to an element in the data structure. For more information about the applicable values (strings) for this variable, see 5.8.1.1.

5.8.1.1 Light-weight relative resource paths

The following table describes the types of light-weight resources that can be accessed by using this resource, applicable methods, and the links to data structures that contain values (strings) for those relative resource paths.

Light-weight resource type	Method supported	Description
Presence attribute groups	GET, PUT, DELETE	Enables access to presence attributes related to Person, Service or Device. See data structure 5.2.3 for possible values for the light-weight relative resource path.
Person attributes	GET, PUT, DELETE	Enables access to a single presence attribute related to a person. See data structure 5.2.4 for possible values for the light-weight relative resource path.

Service attributes	GET, PUT, DELETE	Enables access to a single presence attribute related to a service. See data structure 5.2.5 for possible values for the light-weight relative resource path.
Device attributes	GET, PUT, DELETE	Enables access to a single presence attribute related to a device. See data structure 5.2.6 for possible values for the light-weight relative resource path.

5.8.2 Response Codes

5.8.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.8.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Presence, see Section 6.

5.8.3 GET

This operation is used to retrieve a particular persistent presence attribute for the specified user.

5.8.3.1 Example: retrieving individual persistent presence attribute (Informative)

5.8.3.1.1 Request

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/persistent/person/statusIcon HTTP/1.1
Host: example.com:80
Accept: application/xml
```

5.8.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:statusIcon xmlns:pr="urn:oma:xml:rest:presence:1">
  <statusIconAddress>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/content/pic001.jpg</statusIconAddress>
  <contentType>image/jpg</contentType>
  <eTag>123</eTag>
</pr:statusIcon>
```

5.8.4 PUT

This operation is used to update (or create if it does not exist already) a particular persistent presence attribute for the specified user.

5.8.4.1 Example: updating individual persistent presence attribute (Informative)

5.8.4.1.1 Request

```
PUT /exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/persistent/person/statusIcon HTTP/1.1
Host: example.com:80
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:statusIcon xmlns:pr="urn:oma:xml:rest:presence:1">
  <statusIconAddress>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/content/pic001.jpg</statusIconAddress>
  <contentType>image/jpg</contentType>
  <eTag>456</eTag>
</pr:statusIcon>
```

5.8.4.1.2 Response

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:statusIcon xmlns:pr="urn:oma:xml:rest:presence:1">
  <statusIconAddress>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/content/pic001.jpg</statusIconAddress>
  <contentType>image/jpg</contentType>
  <eTag>456</eTag>
</pr:statusIcon>
```

5.8.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, PUT, DELETE' field in the response as per section 14.7 of [RFC 2616].

5.8.6 DELETE

This operation is used to remove a particular persistent presence attribute.

5.8.6.1 Example: removing individual persistent presence attribute (Informative)

5.8.6.1.1 Request

```
DELETE /exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/persistent/person/mood HTTP/1.1
Host: example.com:80
```

5.8.6.1.2 Response

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

5.9 Resource: Presentity Content list

The resource used is:

http://{serverRoot}/{apiVersion}/presence/{userId}/content

The userId must be percent-encoded according to [RFC3986].

This resource is used to retrieve information about contents stored on the server. The result contains for example the URL for each uploaded content file. The file may consist of an icon/picture etc.

5.9.1 Request URI variables

The following request URI variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com/exampleAPI
apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)
userId	identity of the Presentity that the content is retrieved for. Example: tel:+1-555-100

5.9.2 Response Codes

5.9.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.9.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Presence, see [3GPP 29.199-14].

5.9.3 GET

This operation is used to retrieve a list of content for the specified user.

5.9.3.1 Example: retrieving list of available contents (Informative)

5.9.3.1.1 Request

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-100/content HTTP/1.1
Host: example.com:80
Accept: application/xml
```

5.9.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<pr:contentList xmlns:pr="urn:oma:xml:rest:presence:1">
   <content>
      <link rel="content" href="http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/content/pic003.jpg"/>
      <contentType>image/jpg</contentType>
   </content>
   <content>
      <link rel="content" href="http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/content/pic004.jpg"/>
      <contentType>image/jpg</contentType>
   </content>
  <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/content</resourceURL>
</pr:contentList>
```

5.9.4 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 section 14.7 of [RFC 2616].

5.9.5 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 section 14.7 of [RFC 2616].

5.9.6 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 section 14.7 of [RFC 2616].

5.10 Resource: Individual Presentity content

The resource used is:

http://{serverRoot}/{apiVersion}/presence/{userId}/content/{contentId}

The userId and contentId must be percent-encoded according to [RFC3986].

This resource is used by Presentity to retrieve, upload, and remove an individual content (e.g. an icon). The uploaded content is related to the Presentity by using the "person/status-icon" attribute where the link to the content is stored.

5.10.1 Request URI variables

The following request URI variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com/exampleAPI
apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)

userId	identity of the Presentity that the content is managed for. Example: tel:+1-555-100
contentId	contains an identifier of the content. The identifier may be structured as a relative path consisting of a directory and filename. Example: oma_status-icon/myPicturejpg

5.10.2 Response Codes

5.10.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.10.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Presence, see Section 6.

5.10.3 GET

This operation is used by the Presentity to retrieve the content as specified in the URL.

5.10.3.1 Example: retrieving individual content by Presentity (Informative)

5.10.3.1.1 Request

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-100/content/pic001.jpg HTTP/1.1
Host: example.com:80
Accept: image/jpg
```

5.10.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: image/jpg
Content-Length: nnnn

data
```

5.10.4 PUT

This operation is used to upload new or modify the existing content on the server.

5.10.4.1 Example: uploading/updating individual content by Presentity (Informative)

5.10.4.1.1 Request

```
PUT /exampleAPI/1/presence/tel%3A%2B1-555-100/content/pic001.jpg HTTP/1.1
Host: example.com:80
Content-Type: image/jpg
Content-Length: nnnn

data
```

5.10.4.1.2 Response

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

5.10.5 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 section 14.7 of [RFC 2616].

5.10.6 DELETE

This operation is used to remove the content as specified by in the URL.

5.10.6.1 Example: removing individual content by Presentity (Informative)

5.10.6.1.1 Request

```
DELETE /exampleAPI/1/presence/tel%3A%2B1-555-100/content/pic001.jpg HTTP/1.1
Host: example.com:80
```

5.10.6.1.2 Response

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

5.11 Resource: Watchers list

The resource used is:

http://{serverRoot}/{apiVersion}/presence/{userId}/watchers

The userId must be percent-encoded according to [RFC3986].

This resource is used by the Presentity to retrieve the list of Watchers that are interested in the Presentity's Presence data including the current subscription status. The desired state is provided in a query parameter (e.g. state=pending or state=active).

A typical usage is to retrieve unauthorized users in order to decide whether to allow, block or politely block them.

5.11.1 Request URI variables

The following request URI variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com/exampleAPI
apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)
userId	identity of the Presentity that the Watcher list is retrieved for. Example: tel:+1-555-100

5.11.2 Response Codes

5.11.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.11.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Presence, see [3GPP 29.199-14].

5.11.3 GET

This operation is used to retrieve a list of Watchers (including corresponding subscription status) interested in the Presentity's presence information.

Name	Type/value	Optional	Description
resourceStatusFilter	ResourceStatus [0..unbounded]	Yes	Allows the Presentity to indicate the Watcher resource state it is interested of in the response according to the values specified in section 5.2.59. Example: "?resourceStatusFilter=Pending"

5.11.3.1 Example: retrieving list of Watchers (Informative)

5.11.3.1.1 Request

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-100/watchers?resourceStatusFilter=Pending HTTP/1.1
Host: example.com:80
Accept: application/xml
```

5.11.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<pr:watcherList xmlns:pr="urn:oma:xml:rest:presence:1">
  <watcher>
    <watcherUserId>tel:+1-555-101</watcherUserId>
    <displayName>Bob</displayName>
    <resourceStatus>Pending</resourceStatus>
  </watcher>
  <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/watchers</resourceURL>
</pr:watcherList>
```

5.11.4 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 section 14.7 of [RFC 2616].

5.11.5 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 section 14.7 of [RFC 2616].

5.11.6 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 section 14.7 of [RFC 2616].

5.12 Resource: Individual Watcher

The resource used is:

http://{serverRoot}/{apiVersion}/presence/{userId}/watchers/{watcherUserId}

The userId and watcherUserId must be percent-encoded according to [RFC3986].

This resource is used to retrieve subscription status about bout an individual Watcher.

5.12.1 Request URI variables

The following request URI variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com/exampleAPI
apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)
userId	identity of the Presentity that the Watcher is retrieved for. Example: tel:+1-555-100
watcherUserId	identity of the Watcher to retrieve information about. Example: tel:+1-555-101

5.12.2 Response Codes

5.12.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.12.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Presence, see [3GPP 29.199-14].

5.12.3 GET

This operation is used by a Presentity to retrieve subscription status about an individual Watcher.

5.12.3.1 Example: retrieving individual Watcher (Informative)

5.12.3.1.1 Request

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-100/watchers/{tel%3A%2B1-555-101 HTTP/1.1
```

```
Host: example.com:80
Accept: application/xml
```

5.12.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:watcher xmlns:pr="urn:oma:xml:rest:presence:1">
  <watcherUserId>tel:+1-555-101</watcherUserId>
  <displayName>Bob</displayName>
  <resourceStatus>Pending</resourceStatus>
  <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/watchers/tel%3A%2B1-555-101</resourceURL>
</pr:watcher>
```

5.12.4 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 section 14.7 of [RFC 2616].

5.12.5 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 section 14.7 of [RFC 2616].

5.12.6 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 section 14.7 of [RFC 2616].

5.13 Resource: Authorization rules

The resource used is:

http://{serverRoot}/{apiVersion}/presence/{userId}/authorization/rules

The userId must be percent-encoded according to [RFC3986].

This resource is used by a Presentity to create and retrieve authorization rules. The authorization rules controls who will have access to Presentity's presence information. A Watcher may be authorized to all or a subset of the available presence attributes.

5.13.1 Request URI variables

The following request URI variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com/exampleAPI

apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)
userId	identity of the Presentity that the authorization rules are managed for. Example: tel:+1-555-100

5.13.2 Response Codes

5.13.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.13.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Presence, see Section 6.

5.13.3 GET

This operation is used by a Presentity to retrieve all authorization rules.

5.13.3.1 Example: retrieving all authorization rules (Informative)

5.13.3.1.1 Request

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-100/authorization/rules HTTP/1.1
Host: example.com:80
Accept: application/xml
```

5.13.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:ruleList xmlns:pr="urn:oma:xml:rest:presence:1">
   <rule ruleName="allowList">
     <watcherUserId>tel:+1-555-102</watcherUserId>
     <watcherUserId>tel:+1-555-104</watcherUserId>
     <decision>Allow</decision>
   </rule>
   <rule ruleName="blockList">
     <memberListId>myBlockList</memberListId>
     <decision>Block</decision>
   </rule>
  <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/authorization/rules</resourceURL>
</pr:ruleList>
```

5.13.4 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 section 14.7 of [RFC 2616].

5.13.5 POST

This operation is used by a Presentity to create a new authorization rule.

5.13.5.1 Example: creating an authorization rule (Informative)

5.13.5.1.1 Request

```
POST /exampleAPI/1/presence/tel%3A%2B1-555-100/authorization/rules HTTP/1.1
Host: example.com:80
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:rule ruleName="otherUsers" xmlns:pr="urn:oma:xml:rest:presence:1">
  <otherUser/>
  <decision>Confirm</decision>
</pr:rule>
```

5.13.5.1.2 Response

```
HTTP/1.1 201 Created
Location: http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/authorization/rules/rule003
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:rule ruleName="otherUsers" xmlns:pr="urn:oma:xml:rest:presence:1">
  <otherUser/>
  <decision>Confirm</decision>
  <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/authorization/rules/rule003</resourceURL>
</pr:rule>
```

5.13.5.2 Example 2: creating an authorization rule, response with resourceReference (Informative)

5.13.5.2.1 Request

```
POST /exampleAPI/1/presence/tel%3A%2B1-555-100/authorization/rules HTTP/1.1
Host: example.com:80
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:rule ruleName="otherUsers" xmlns:pr="urn:oma:xml:rest:presence:1">
  <otherUser/>
  <decision>Confirm</decision>
</pr:rule>
```

5.13.5.2.2 Response

```
HTTP/1.1 201 Created
Location: http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/authorization/rules/rule003
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<common:resourceReference xmlns:common="urn:oma:xml:rest:common:1">
  <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/authorization/rules/rule003</resourceURL>
</common:resourceReference>
```

5.13.6 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 section 14.7 of [RFC 2616].

5.14 Resource: Individual authorization rule

The resource used is:

http://{serverRoot}/{apiVersion}/presence/{userId}/authorization/rules/{ruleId}

The userId and ruleId must be percent-encoded according to [RFC3986].

This resource is used by a Presentity to manage an individual authorization rule.

5.14.1 Request URI variables

The following request URI variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com/exampleAPI
apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)
userId	identity of the Presentity that the individual authorization rule is managed for. Example: tel:+1-555-100
ruleId	identity of the rule generated by the system

5.14.2 Response Codes

5.14.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.14.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Presence, see Section 6.

5.14.3 GET

This operation is used by a Presentity to retrieve an authorization rule.

5.14.3.1 Example: retrieving an authorization rule (Informative)

5.14.3.1.1 Request

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-100/authorization/rules/rule001 HTTP/1.1
Host: example.com:80
Accept: application/xml
```

5.14.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:rule ruleName="allowList" xmlns:pr="urn:oma:xml:rest:presence:1">
  <watcherUserId>tel:+1-555-102</watcherUserId>
  <watcherUserId>tel:+1-555-104</watcherUserId>
  <decision>Allow</decision>
  <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/authorization/rules/rule001</resourceURL>
</pr:rule>
```

5.14.4 PUT

This operation is used by a Presentity to update an authorization rule.

5.14.4.1 Example: updating an authorisation rule (Informative)

5.14.4.1.1 Request

```
PUT /exampleAPI/1/presence/tel%3A%2B1-555-100/authorization/rules/rule001 HTTP/1.1
Host: example.com:80
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:rule ruleName="allowList" xmlns:pr="urn:oma:xml:rest:presence:1">
  <watcherUserId>tel:+1-555-102</watcherUserId>
  <watcherUserId>tel:+1-555-104</watcherUserId>
  <watcherUserId>tel:+1-555-105</watcherUserId>
  <decision>Allow</decision>
</pr:rule>
```

5.14.4.1.2 Response

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:rule ruleName="allowList" xmlns:pr="urn:oma:xml:rest:presence:1">
  <watcherUserId>tel:+1-555-102</watcherUserId>
  <watcherUserId>tel:+1-555-104</watcherUserId>
  <watcherUserId>tel:+1-555-105</watcherUserId>
  <decision>Allow</decision>
  <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/authorization/rules/rule001</resourceURL>
</pr:rule>
```

5.14.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, PUT, DELETE' field in the response as per section 14.7 of [RFC 2616].

5.14.6 DELETE

This operation is used by a Presentity to remove an authorization rule.

5.14.6.1 Example: removing an authorisation rule (Informative)

5.14.6.1.1 Request

```
DELETE /exampleAPI/1/presence/tel%3A%2B1-555-100/authorization/rules/rule002 HTTP/1.1
Host: example.com:80
```

5.14.6.1.2 Response

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

5.15 Resource: Individual authorization rule data

The resource used is:

http://{serverRoot}/{apiVersion}/presence/{userId}/authorization/rules/{ruleId}/[ResourceRelPath]

The userId, ruleId and ResourceRelPath must be percent-encoded according to [RFC3986].

This resource is used by Presentity to update an authorization rule by specifying the identity to authorize in the URL. Users, member lists or domains may be authorized using this operation. The resource URL consists of heavy-weight resource path, **http://{serverRoot}/{apiVersion}/presence/{userId}/authorization/rules/{ruleId}**/, and an extension of the resource URL path, which is relative resource path for a light-weight resource, and it is represented by [**ResourceRelPath].**

5.15.1 Request URI variables

The following request URI variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com/exampleAPI
apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)
userId	identity of the Presentity that the authorization rule data is managed for. Example: tel:+1-555-100
ruleId	identifier of the rule generated by the system
[ResourceRelPath]	Relative resource path for a light-weight resource, consisting of a relative path down to an element in the data structure. For more information about the applicable values (strings) for this variable, see 5.15.1.1.

5.15.1.1 Light-weight relative resource paths

The following table describes the type of light-weight resources that can be accessed by using this resource, applicable methods, and the link to a data structure that contains values (strings) for those relative resource paths.

Light-weight resource type	Method supported	Description
Watcher identities	GET, PUT, DELETE	Enables access to authorization data related to a specific authorization rule. See data structure 5.2.13 for possible values for the light-weight relative resource path.

5.15.2 Response Codes

5.15.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.15.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Presence, see Section 6.

5.15.3 GET

This operation is used by a Presentity to retrieve a user, member list or domain from an authorization rule.

5.15.3.1 Example: retrieving individual authorization rule data (Informative)

5.15.3.1.1 Request

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-100/authorization/rules/rule002/watchers/tel%3A%2B1-555-102 HTTP/1.1
Host: example.com:80
Accept: application/xml
```

5.15.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<pr:watcherUserId xmlns:pr="urn:oma:xml:rest:presence:1">
   tel:+1-555-102
</pr:watcherUserId>
```

5.15.4 PUT

This operation is used by a Presentity to authorize a user, member list or domain by including its identity in the request.

5.15.4.1 Example: updating individual authorization rule data (Informative)

5.15.4.1.1 Request

```
PUT /exampleAPI/1/presence/tel%3A%2B1-555-100/authorization/rules/rule002/watchers/tel%3A%2B1-555-103 HTTP/1.1
Host: example.com:80
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:watcherUserId xmlns:pr="urn:oma:xml:rest:presence:1">
     tel:+1-555-103
</pr:watcherUserId>
```

5.15.4.1.2 Response

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:watcherUserId xmlns:pr="urn:oma:xml:rest:presence:1">
    tel:+1-555-103
</pr:watcherUserId>
```

5.15.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, PUT, DELETE' field in the response as per section 14.7 of [RFC 2616].

5.15.6 DELETE

This operation is used by a Presentity to remove a user, member list or domain from an authorization rule.

5.15.6.1 Example: removing individual authorization rule data (Informative)

5.15.6.1.1 Request

```
DELETE /exampleAPI/1/presence/tel%3A%2B1-555-100/authorization/rules/rule002/watchers/tel%3A%2B1-555-103
Host: example.com:80
```

5.15.6.1.2 Response

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

5.16 Resource: Presence information by Watcher

The resource used is:

http://{serverRoot}/{apiVersion}/presence/{userId}/presenceContacts/{presentityUserId}

The userId and presentityUserId must be percent-encoded according to [RFC3986].

This resource is used by a Watcher to retrieve presence information about a single Presentity.

5.16.1 Request URI variables

The following request URI variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com/exampleAPI
apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)
userId	identity of the Watcher retrieving the presence data from the Presentity. Example: tel:+1-555-101
presentityUserId	identity of the Presentity owning the presence data Example: tel:+1-555-100

5.16.2 Response Codes

5.16.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.16.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Presence, see [3GPP 29.199-14].

5.16.3 GET

This operation is used by a Watcher to retrieve presence information about Presentity.

Request URL parameters are:

Name	Type/value	Optional	Description
presenceFilter	xsd:anyURI [0..unbounded]	Yes	Allows the Watcher to indicate what presence data it is interested of. The desired attributes are indicated with relative paths according to the [ResourceRelPath] in sections 5.2.3, 5.2.4, 5.2.5 and 5.2.6 with the following clarifications: The 'serviceId', 'version' and 'deviceId' MAY be specified using a "*" meaning that the filter applies to several services and devices respectively Example: “?presenceFilter=person/mood&presenceFilter=service/*/*/icon”
anonymous	empty	Yes	Allows the Watcher to request that its user identity is not revealed to the Presentity. Example: ?anonymous

5.16.3.1 Example 1: retrieving all presence data for Presentity (Informative)

5.16.3.1.1 Request

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-101/presenceContacts/tel%3A%2B1-555-100 HTTP/1.1
Host: example.com:80
Accept: application/xml
```

5.16.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:presenceContact xmlns:pr="urn:oma:xml:rest:presence:1">
 <presentityUserId>tel:+1-555-100</presentityUserId>
 <presence>
  <person>
   <mood>
    <moodValue>Happy</moodValue>
   </mood>
   <noteList>
    <note xml:lang="en">Im on vacation!</note>
   </noteList>
  </person>
  <service serviceId="org.openmobilealliance:IM-Session" version="1.0">
   <serviceAvailability>Open</serviceAvailability>
   <devices>
    <deviceId>mac:321</deviceId>
   </devices>
  </service>
  <device deviceId="mac:321">
     <networkAvailability>
      <network id="GPRS">
       <connectionStatus>Active</connectionStatus>
      </network>
```

```
      </networkAvailability>
    </device>
  </presence>
  <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/presenceContacts/tel%3A%2B1-555-
100</resourceURL>
</pr:presenceContact>
```

5.16.3.2 Example 2: retrieving presence for Presentity by using filter (Informative)

5.16.3.2.1 Request

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-101/presenceContacts/tel%3A%2B1-555-
100?presenceFilter=person/mood&presenceFilter=service/org.openmobilealliance:IM-Session/1.0/serviceAvailability HTTP/1.1
Host: example.com:80
Accept: application/xml
```

5.16.3.2.2 Response

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:presenceContact xmlns:pr="urn:oma:xml:rest:presence:1">
  <presentityUserId>tel:+1-555-100</presentityUserId>
  <presence>
    <person>
      <mood>
        <moodValue>Happy</moodValue>
      </mood>
    </person>
    <service serviceId="org.openmobilealliance:IM-Session" version="1.0">
      <serviceAvailability>Open</serviceAvailability>
      <devices>
        <deviceId>mac:321</deviceId>
      </devices>
    </service>
  </presence>
  <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/presenceContacts/tel%3A%2B1-555-
100</resourceURL>
</pr:presenceContact>
```

5.16.4 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 section 14.7 of [RFC 2616].

5.16.5 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 section 14.7 of [RFC 2616].

5.16.6 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 section 14.7 of [RFC 2616].

5.17 Resource: Individual presence attribute by Watcher

The resource used is:

http://{serverRoot}/{apiVersion}/presence/{userId}/presenceContacts/{presentityUserId}/[ResourceRelPath]

The userId, presentityUserId and ResourceRelPath must be percent-encoded according to [RFC3986].

This resource is used by a Watcher to retrieve an individual presence attribute from a single Presentity. The resource URL consists of heavy-weight resource path, **http://{serverRoot}/{apiVersion}/presence/{userId}/presenceContacts/{presentityUserId}/,** and an extension of the resource URL path, which is relative resource path for a light-weight resource, and it is represented by [**ResourceRelPath].**

5.17.1 Request URI variables

The following request URI variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com/exampleAPI
apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)
userId	identity of the Watcher retrieving the presence data from the Presentity. Example: tel:+1-555-101
presentityUserId	identity of the Presentity owning the presence data Example: tel:+1-555-100
[ResourceRelPath]	Relative resource path for a light-weight resource, consisting of a relative path down to an element in the data structure. For more information about the applicable values (strings) for this variable, see 5.17.1.1.

5.17.1.1 Light-weight relative resource paths

The following table describes the types of light-weight resources that can be accessed using this resource, applicable methods, and the links to data structures that contain values (strings) for those relative resource paths.

Light-weight resource type	Method supported	Description
Presence attribute groups	GET, PUT, DELETE	Enables access to presence attributes related to Person, Service or Device. See data structure 5.2.3 for possible values for the

		light-weight relative resource path.
Person attributes	GET, PUT, DELETE	Enables access to a single presence attribute related to a person. See data structure 5.2.4 for possible values for the light-weight relative resource path.
Service attributes	GET, PUT, DELETE	Enables access to a single presence attribute related to a service. See data structure 5.2.5 for possible values for the light-weight relative resource path.
Device attributes	GET, PUT, DELETE	Enables access to a single presence attribute related to a device. See data structure 5.2.6 for possible values for the light-weight relative resource path.

5.17.2 Response Codes

5.17.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.17.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Presence, see [3GPP 29.199-14].

5.17.3 GET

This operation is used by a Watcher to retrieve presence information about Presentity.

Request URL parameters are:

Name	Type/value	Optional	Description
anonymous	empty	Yes	Allows the Watcher to request that its user identity is not revealed to the Presentity. Example: ?anonymous

5.17.3.1 Example: retrieving individual presence attribute for Presentity (Informative)

5.17.3.1.1 Request

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-101/presenceContacts/tel%3A%2B1-555-100/person/noteList HTTP/1.1
Host: example.com:80
Accept: application/xml
```

5.17.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:presenceContact xmlns:pr="urn:oma:xml:rest:presence:1">
  <presentityUserId>tel:+1-555-100</presentityUserId>
  <presence>
    <person>
      <noteList>
        <note xml:lang="en">Im on vacation!</note>
      </noteList>
    </person>
  </presence>
</pr:presenceContact>
```

5.17.4 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 section 14.7 of [RFC 2616].

5.17.5 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 section 14.7 of [RFC 2616].

5.17.6 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 section 14.7 of [RFC 2616].

5.18 Resource: Presence information by Watcher for a member list

The resource used is:

http://{serverRoot}/{apiVersion}/presence/{userId}/presenceLists/{presenceListId}

The userId and presenceListId must be percent-encoded according to [RFC3986].

This resource is used by a Watcher to retrieve presence information about Presentities in a Presence list.

5.18.1 Request URI variables

The following request URI variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com/exampleAPI
apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)

userId	identity of the Watcher retrieving the presence data from the Presence list. Example: tel:+1-555-101
presenceListId	identity of the Presence list Example: myFriends

5.18.2 Response Codes

5.18.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.18.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Presence, see [3GPP 29.199-14].

5.18.3 GET

This operation is used by a Watcher to retrieve presence information about Presentities in a Presence list.

Request URL parameters are:

Name	Type/value	Optional	Description
presenceFilter	xsd:anyURI [0..unbounded]	Yes	Allows the Watcher to indicate what presence data it is interested of. The desired attributes are indicated with resource relative paths according to the [ResourceRelPath] in sections 5.2.3, 5.2.4, 5.2.5 and 5.2.6 with the following clarifications: The 'serviceId', 'version' and 'deviceId' MAY be specified using a "*" meaning that the filter applies to several services and devices respectively. Example: “?presenceFilter=person/mood&presenceFilter=service/*/*/icon”
anonymous	empty	Yes	Allows the Watcher to request that its user identity is not revealed to the Presentity. Example: ?anonymous

5.18.3.1 Example: retrieving presence data for all Presentities in a Presence list (Informative)

5.18.3.1.1 Request

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-101/presenceLists/myFriends HTTP/1.1
Host: example.com:80
Accept: application/xml
```

5.18.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn
```

```
<?xml version="1.0" encoding="UTF-8"?>
<pr:presenceList xmlns:pr="urn:oma:xml:rest:presence:1">
 <presenceContact>
  <presentityUserId>tel:+1-555-100</presentityUserId>
  <resourceStatus>Active</resourceStatus>
  <presence>
   <person>
    <mood>
     <moodValue>Happy</moodValue>
    </mood>
    <noteList>
     <note xml:lang="en">Im on vacation!</note>
    </noteList>
   </person>
   <service serviceId="org.openmobilealliance:IM-Session" version="1.0">
    <serviceAvailability>Open</serviceAvailability>
    <devices>
     <deviceId>mac:321</deviceId>
    </devices>
   </service>
   <device deviceId="mac:321">
    <networkAvailability>
     <network id="GPRS">
      <connectionStatus>Active</connectionStatus>
     </network>
    </networkAvailability>
   </device>
  </presence>
 </presenceContact>
 <presenceContact>
  <presentityUserId>tel:+1-555-102</presentityUserId>
  <resourceStatus>Pending</resourceStatus>
 </presenceContact>
 <presenceContact>
  <presentityUserId>tel:+1-555-104</presentityUserId>
  <resourceStatus>TerminatedNoResource</resourceStatus>
 </presenceContact>
 <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/presenceLists/myFriends</resourceURL>
</pr:presenceList>
```

5.18.4 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 section 14.7 of [RFC 2616].

5.18.5 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 section 14.7 of [RFC 2616].

5.18.6 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 section 14.7 of [RFC 2616].

5.19 Resource: Content by Watcher

The resource used is:

http://{serverRoot}/{apiVersion}/presence/{userId}/PresenceContactsContent/{presentityUserId}/{contentId}

The userId, presentityUserId and contentId must be percent-encoded according to [RFC3986].

This resource is used by a Watcher to retrieve content from Presentity.

The Watcher is only allowed to retrieve it if has been authorized by the Presentity.

5.19.1 Request URI variables

The following request URI variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com/exampleAPI
apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)
userId	identity of the Watcher retrieving the content data from the Presentity. Example: tel:+1-555-101
presentityUserId	identity of the Presentity owning the content data Example: tel:+1-555-100

5.19.2 Response Codes

5.19.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.19.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Presence, see [3GPP 29.199-14].

5.19.3 GET

This operation is used by a Watcher to retrieve content, such as a picture/icon from Presentity.

5.19.3.1 Example: retrieving content by Watcher (Informative)

5.19.3.1.1 Request

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-101/presenceContactsContent/tel%3A%2B1-555-100/pic001.jpg HTTP/1.1
Host: example.com:80
Accept: image/jpg
```

5.19.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: image/jpg
Content-Length: nnnn

data
```

5.19.4 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 section 14.7 of [RFC 2616].

5.19.5 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 section 14.7 of [RFC 2616].

5.19.6 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 section 14.7 of [RFC 2616].

5.20 Resource: All subscriptions

The resource used is:

http://{serverRoot}/{apiVersion}/presence/{userId}/subscriptions

The userId must be percent-encoded according to [RFC3986].

This resource is used to retrieve all subscriptions that the user has created. It includes all active Presence-, Presence list- and Watchers subscriptions.

5.20.1 Request URI variables

The following request URI variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com/exampleAPI
apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)
userId	identity of the user retrieving the subscriptions. Example: tel:+1-555-101

5.20.2 Response Codes

5.20.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.20.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Presence, see [3GPP 29.199-14].

5.20.3 GET

This operation is used by a user to retrieve all active subscriptions.

5.20.3.1 Example: retrieving all active subscriptions for user (Informative)

5.20.3.1.1 Request

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions HTTP/1.1
Host: example.com:80
Accept: application/xml
```

5.20.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:subscriptionList xmlns:pr="urn:oma:xml:rest:presence:1">
 <presenceSubscriptionList>
  <presenceSubscription>
   <presentityUserId>tel:+1-555-100</presentityUserId>
   <callbackReference>
    <notifyURL>http://application.example.com/notifications/presenceNotification</notifyURL>
    <callbackData>1234</callbackData>
   </callbackReference>
   <clientCorrelator>321</clientCorrelator>
   <applicationTag>myApp</applicationTag>
   <duration>5246</duration>
   <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-
101/subscriptions/presenceSubscriptions/tel%3A%2B1-555-100/sub001
   </resourceURL>
  </presenceSubscription>
  <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-
101/subscriptions/presenceSubscriptions</resourceURL>
 </presenceSubscriptionList>
 <presenceListSubscriptionCollection>
  <presenceListSubscription>
   <presenceListId>myFriends</presenceListId>
   <callbackReference>
    <notifyURL>http://application.example.com/notifications/presenceListNotification</notifyURL>
    <callbackData>2345</callbackData>
   </callbackReference>
   <clientCorrelator>432</clientCorrelator>
   <applicationTag>myApp</applicationTag>
   <duration>4629</duration>
   <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-
101/subscriptions/presenceListSubscriptions/myFriends/
   sub002</resourceURL>
```

```
   </presenceListSubscription>
   <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-
101/subscriptions/presenceListSubscriptions</resourceURL>
  </presenceListSubscriptionCollection>
  <watchersSubscriptionList>
   <watcherSubscription>
    <presentityUserId>tel:+1-555-100</presentityUserId>
    <callbackReference>
     <notifyURL>http://application.example.com/notifications/watchersNotification</notifyURL>
     <callbackData>3456</callbackData>
    </callbackReference>
    <clientCorrelator>543</clientCorrelator>
    <applicationTag>myApp</applicationTag>
    <duration>2413</duration>
    <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/watchersSubscriptions/sub003
    </resourceURL>
   </watcherSubscription>
   <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-
101/subscriptions/watchersSubscriptions</resourceURL>
  </watchersSubscriptionList>
  <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions</resourceURL>
</pr:subscriptionList>
```

5.20.4 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 section 14.7 of [RFC 2616].

5.20.5 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 section 14.7 of [RFC 2616].

5.20.6 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 section 14.7 of [RFC 2616].

5.21 Resource: All Watchers subscriptions

The resource used is:

http://{serverRoot}/{apiVersion}/presence/{userId}/subscriptions/watchersSubscriptions

The userId must be percent-encoded according to [RFC3986].

This resource is used by a Presentity to manage Watchers subscriptions, i.e. subscriptions for changes in the Watchers list. The list contains Watchers that are subscribing for presence information about the Presentity.

For instance, a notification will be generated when there is a new Watcher for Presentity and the authorization decision evaluates to 'Confirm'.

5.21.1 Request URI variables

The following request URI variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com/exampleAPI
apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)
userId	identity of the Presentity retrieving the subscriptions. Example: tel:+1-555-100

5.21.2 Response Codes

5.21.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.21.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Presence, see Section 6.

5.21.3 GET

This operation is used by a Presentity to retrieve all its active Watchers subscriptions.

5.21.3.1 Example: retrieving all Watchers subscriptions (Informative)

5.21.3.1.1 Request

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-100/subscriptions/watchersSubscriptions HTTP/1.1
Host: example.com:80
Accept: application/xml
```

5.21.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<pr:watchersSubscriptionList xmlns:pr="urn:oma:xml:rest:presence:1">
  <watcherSubscription>
    <presentityUserId>tel:+1-555-100</presentityUserId>
    <callbackReference>
      <notifyURL>http://application.example.com/notifications/watchersNotification</notifyURL>
      <callbackData>1234</callbackData>
    </callbackReference>
    <clientCorrelator>321</clientCorrelator>
    <applicationTag>myApp</applicationTag>
    <duration>5246</duration>
    <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/subscriptions/watchersSubscriptions/tel%3A%2B1-
```

```
555-100/sub001
   </resourceURL>
  </watcherSubscription>
  <watcherSubscription>
   <presentityUserId>tel:+1-555-100</presentityUserId>
   <callbackReference>
    <notifyURL>http://application.example.com/notifications/watchersNotification</notifyURL>
    <callbackData>4321</callbackData>
   </callbackReference>
   <clientCorrelator>123</clientCorrelator>
   <applicationTag>myApp</applicationTag>
   <duration>5237</duration>
   <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/subscriptions/watchersSubscriptions/tel%3A%2B1-
555-100/sub002
   </resourceURL>
  </watcherSubscription>
  <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-
100/subscriptions/watchersSubscriptions</resourceURL>
</pr:watchersSubscriptionList>
```

5.21.4 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 section 14.7 of [RFC 2616].

5.21.5 POST

This operation is used by a Presentity to create a new Watchers subscription.

5.21.5.1 Example: creating new Watchers subscription (Informative)

5.21.5.1.1 Request

```
POST /exampleAPI/1/presence/tel%3A%2B1-555-100/subscriptions/watchersSubscriptions HTTP/1.1
Host: example.com:80
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:watcherSubscription xmlns:pr="urn:oma:xml:rest:presence:1">
  <presentityUserId>tel:+1-555-100</presentityUserId>
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/watchersNotification</notifyURL>
    <callbackData>1234</callbackData>
  </callbackReference>
  <clientCorrelator>321</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <duration>7200</duration>
</pr:watcherSubscription>
```

5.21.5.1.2 Response

```
HTTP/1.1 201 Created
Location: http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/subscriptions/watchersSubscriptions/sub001
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:watcherSubscription xmlns:pr="urn:oma:xml:rest:presence:1">
  <presentityUserId>tel:+1-555-100</presentityUserId>
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/watchersNotification</notifyURL>
    <callbackData>1234</callbackData>
  </callbackReference>
  <clientCorrelator>321</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <duration>3600</duration>
  <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/subscriptions/watchersSubscriptions/sub001
  </resourceURL>
</pr:watcherSubscription>
```

In the above example, the Presentity requested the duration of the subscription to be 7200 seconds. However the server policy for that particular server allows a maximum of 3600 seconds for the duration of a subscription, which is reflected in the response.

5.21.6 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 section 14.7 of [RFC 2616].

5.22 Resource: Individual Watchers subscription

The resource used is:

http://{serverRoot}/{apiVersion}/presence/{userId}/subscriptions/watchersSubscriptions/{subscriptionId}

The userId and subscriptionId must be percent-encoded according to [RFC3986].

This resource is used by a Presentity to manage an individual Watchers subscription.

5.22.1 Request URI variables

The following request URI variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com/exampleAPI
apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)
userId	identity of the Presentity managing the subscription. Example: tel:+1-555-100
subscriptionId	identifier of the subscription generated by the system

5.22.2 Response Codes

5.22.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.22.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Presence, see Section 6.

5.22.3 GET

This operation is used by a Presentity to retrieve an individual Watchers subscription.

5.22.3.1 Example: retrieving individual Watchers subscription (Informative)

5.22.3.1.1 Request

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-100/subscriptions/watchersSubscriptions/sub001 HTTP/1.1
Host: example.com:80
Accept: application/xml
```

5.22.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:watcherSubscription xmlns:pr="urn:oma:xml:rest:presence:1">
  <presentityUserId>tel:+1-555-100</presentityUserId>
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/watchersNotification</notifyURL>
    <callbackData>1234</callbackData>
  </callbackReference>
  <clientCorrelator>321</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <duration>5246</duration>
  <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/subscriptions/watchersSubscriptions/sub001
  </resourceURL>
</pr:watcherSubscription>
```

5.22.4 PUT

This operation is used by a Presentity to update subscription parameters (e.g. duration, filter, frequency etc) for an ongoing Watchers subscription.

5.22.4.1 Example: updating individual Watchers subscription (Informative)

5.22.4.1.1 Request

```
PUT /exampleAPI/1/presence/tel%3A%2B1-555-100/subscriptions/watchersSubscriptions/sub001 HTTP/1.1
```

```
Host: example.com:80
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:watcherSubscription xmlns:pr="urn:oma:xml:rest:presence:1">
  <presentityUserId>tel:+1-555-100</presentityUserId>
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/watchersNotification</notifyURL>
    <callbackData>1234</callbackData>
  </callbackReference>
  <clientCorrelator>321</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <duration>7200</duration>
</pr:watcherSubscription>
```

5.22.4.1.2 Response

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:watcherSubscription xmlns:pr="urn:oma:xml:rest:presence:1">
  <presentityUserId>tel:+1-555-100</presentityUserId>
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/watchersNotification</notifyURL>
    <callbackData>1234</callbackData>
  </callbackReference>
  <clientCorrelator>321</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <duration>7200</duration>
  <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/subscriptions/watchersSubscriptions/sub001
  </resourceURL>
</pr:watcherSubscription>
```

5.22.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, PUT, DELETE' field in the response as per section 14.7 of [RFC 2616].

5.22.6 DELETE

This operation is used by a Presentity to terminate an active Watchers subscription.

5.22.6.1 Example: terminating individual Watchers subscription (Informative)

5.22.6.1.1 Request

```
DELETE /exampleAPI/1/presence/tel%3A%2B1-555-100/subscriptions/watchersSubscriptions/sub001 HTTP/1.1
Host: example.com:80
```

5.22.6.1.2 Response

HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT

5.23 Resource: Watchers notification

The resource URL is provided by the Presentity client when the subscription was created. ParlayREST does not make any assumption about the structure of this URL.

A notification is generated by the system in the following occasions:

Type of notification	Generated in the following occasions:
Initial notification	The subscription was successfully created
Subsequent notification	A change in the Watchers list. Please note that a request to extend the duration of a subscription does not generate a new notification.
Final notification	An 'Active' subscription where the Presentity was removed from the system. An 'Active' subscription that was terminated for an unknown reason.

5.23.1 Request URI variables

Provided by the Presentity client

5.23.2 Response Codes

5.23.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.23.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Presence, see [3GPP 29.199-14].

5.23.3 GET

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, PUT, DELETE' field in the response as per section 14.7 of [RFC 2616].

5.23.4 PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, PUT, DELETE' field in the response as per section 14.7 of [RFC 2616].

5.23.5 POST

This operation is used by the system when a new notification is generated.

5.23.5.1 Example 1: notifying Presentity about change in Watchers status (Informative)

5.23.5.1.1 Request

```
POST /notifications/watchersNotification HTTP/1.1
Host: example.com:80
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:watcherNotification xmlns:pr="urn:oma:xml:rest:presence:1">
  <presentityUserId>tel:+1-555-100</presentityUserId>
  <callbackData>1234</callbackData>
  <resourceStatus>Active</resourceStatus>
  <watcherList>
    <watcher>
      <watcherUserId>tel:+1-555-101</watcherUserId>
      <displayName>Bob</displayName>
      <resourceStatus>Pending</resourceStatus>
    </watcher>
  </watcherList>
  <link rel="WatcherSubscription"
   href="http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/subscriptions/watchersSubscriptions/sub001"/>
</pr:watcherNotification>
```

5.23.5.1.2 Response

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

5.23.5.2 Example2: notifying Watcher about subscription time out (Informative)

5.23.5.2.1 Request

```
POST /notifications/watchersNotification HTTP/1.1
Host: example.com:80
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:watcherNotification xmlns:pr="urn:oma:xml:rest:presence:1">
  <presentityUserId>tel:+1-555-100</presentityUserId>
  <callbackData>1234</callbackData>
  <resourceStatus>TerminatedTimeout</resourceStatus>
<link rel="WatcherSubscription"
 href="http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/subscriptions/watchersSubscriptions/sub001"/>
</pr:watcherNotification>
```

5.23.5.2.2 Response

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

5.23.5.3 Example3: notifying Watcher about termination of Watchers subscription (reason unknown) (Informative)

5.23.5.3.1 Request

```
POST /notifications/watchersNotification HTTP/1.1
Host: example.com:80
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:watcherNotification xmlns:pr="urn:oma:xml:rest:presence:1">
  <presentityUserId>tel:+1-555-100</presentityUserId>
  <callbackData>1234</callbackData>
  <resourceStatus>TerminatedOther</resourceStatus>
<link rel="WatcherSubscription"
 href="http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/subscriptions/watchersSubscriptions/sub001"/>
</pr:watcherNotification>
```

5.23.5.3.2 Response

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

5.23.6 DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, PUT, DELETE' field in the response as per section 14.7 of [RFC 2616].

5.24 Resource: All Presence subscriptions

The resource used is:

http://{serverRoot}/{apiVersion}/presence/{userId}/subscriptions/presenceSubscriptions

The userId must be percent-encoded according to [RFC3986].

This resource is used to retrieve all active Presence subscriptions for all Presentities.

5.24.1 Request URI variables

The following request URI variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com/exampleAPI

apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)
userId	identity of the Watcher retrieving the subscriptions. Example: tel:+1-555-101

5.24.2 Response Codes

5.24.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.24.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Presence, see [3GPP 29.199-14].

5.24.3 GET

This operation is used by a Watcher to retrieve all active Presence subscriptions for all Presentities.

5.24.3.1 Example: retrieving all Presence subscriptions for all Presentities (Informative)

5.24.3.1.1 Request

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceSubscriptions HTTP/1.1
Host: example.com:80
Accept: application/xml
```

5.24.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:presenceSubscriptionList xmlns:pr="urn:oma:xml:rest:presence:1">
  <presenceSubscription>
    <presentityUserId>tel:+1-555-100</presentityUserId>
    <callbackReference>
      <notifyURL>http://application.example.com/notifications/presenceNotification</notifyURL>
      <callbackData>1234</callbackData>
    </callbackReference>
    <clientCorrelator>321</clientCorrelator>
    <applicationTag>myApp</applicationTag>
    <duration>5246</duration>
    <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceSubscriptions/tel%3A%2B1-
555-100/sub001
    </resourceURL>
  </presenceSubscription>
  <presenceSubscription>
    <presentityUserId>tel:+1-555-100</presentityUserId>
    <callbackReference>
```

```
    <notifyURL>http://application.example.com/notifications/presenceNotification</notifyURL>
    <callbackData>4321</callbackData>
   </callbackReference>
   <clientCorrelator>123</clientCorrelator>
   <applicationTag>myApp</applicationTag>
   <duration>5237</duration>
   <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceSubscriptions/tel%3A%2B1-
555-101/sub004
   </resourceURL>
  </presenceSubscription>
  <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-
101/subscriptions/presenceSubscriptions</resourceURL>
</pr:presenceSubscriptionList>
```

5.24.4 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 section 14.7 of [RFC 2616].

5.24.5 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 section 14.7 of [RFC 2616].

5.24.6 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 section 14.7 of [RFC 2616].

5.25 Resource: Presence subscriptions

The resource used is:

http://{serverRoot}/{apiVersion}/presence/{userId}/subscriptions/presenceSubscriptions/{presentityUserId}

The userId and presentityUserId must be percent-encoded according to [RFC3986].

This resource is used by a Watcher to manage Presence subscriptions.

5.25.1 Request URI variables

The following request URI variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com/exampleAPI
apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)
userId	identity of the Watcher managing the subscriptions. Example: tel:+1-555-101
presentityUserId	identity of the Presentity owning the presence information Example: tel:+1-555-100

5.25.2 Response Codes

5.25.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.25.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Presence, see Section 6.

5.25.3 GET

This operation is used by a Watcher to retrieve all active Presence subscriptions for the specified Presentity.

5.25.3.1 Example: retrieving all Presence subscriptions for Presentity(Informative)

5.25.3.1.1 Request

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceSubscriptions/tel%3A%2B1-555-100 HTTP/1.1
Host: example.com:80
Accept: application/xml
```

5.25.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<pr:presenceSubscriptionList xmlns:pr="urn:oma:xml:rest:presence:1">
  <presenceSubscription>
    <presentityUserId>tel:+1-555-100</presentityUserId>
    <callbackReference>
      <notifyURL>http://application.example.com/notifications/presenceNotification</notifyURL>
      <callbackData>1234</callbackData>
    </callbackReference>
    <clientCorrelator>321</clientCorrelator>
    <applicationTag>myApp</applicationTag>
    <duration>5246</duration>
    <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceSubscriptions/tel%3A%2B1-
555-100/sub001
    </resourceURL>
  </presenceSubscription>
  <presenceSubscription>
    <presentityUserId>tel:+1-555-100</presentityUserId>
    <callbackReference>
      <notifyURL>http://application2.example.com/notifications/presenceNotification</notifyURL>
      <callbackData>6789</callbackData>
    </callbackReference>
    <clientCorrelator>987</clientCorrelator>
    <applicationTag>myApp</applicationTag>
    <duration>4132</duration>
    <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceSubscriptions/tel%3A%2B1-
555-100/sub005
```

```
   </resourceURL>
  </presenceSubscription>
  <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceSubscriptions/tel%3A%2B1-555-100</resourceURL>
</pr:presenceSubscriptionList>
```

5.25.4 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 section 14.7 of [RFC 2616].

5.25.5 POST

This operation is used by a Watcher to create a new Presence subscription towards the specified Presentity.

5.25.5.1 Example 1: creating new Presence subscription for Presentity(Informative)

5.25.5.1.1 Request

```
POST /exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceSubscriptions/tel%3A%2B1-555-100 HTTP/1.1
Host: example.com:80
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:presenceSubscription xmlns:pr="urn:oma:xml:rest:presence:1">
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/presenceNotification</notifyURL>
    <callbackData>1234</callbackData>
  </callbackReference>
  <clientCorrelator>321</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <duration>7200</duration>
  <frequency>600</frequency>
</pr:presenceSubscription>
```

5.25.5.1.2 Response

```
HTTP/1.1 201 Created
Location: http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/presenceSubscriptions/tel%3A%2B1-555-100/sub001
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<pr:presenceSubscription xmlns:pr="urn:oma:xml:rest:presence:1">
  <presentityUserId>tel:+1-555-100</presentityUserId>
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/presenceNotification</notifyURL>
    <callbackData>1234</callbackData>
  </callbackReference>
  <clientCorrelator>321</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <duration>7200</duration>
```

```
  <frequency>600</frequency>
  <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceSubscriptions/tel%3A%2B1-555-100/sub001
  </resourceURL>
</pr:presenceSubscription>
```

5.25.5.2 Example 2: creating new Presence subscription for unknown Presentity (Informative)

5.25.5.2.1 Request

```
POST /exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceSubscriptions/tel%3A%2B1-555-100 HTTP/1.1
Host: example.com:80
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:presenceSubscription xmlns:pr="urn:oma:xml:rest:presence:1">
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/presenceNotification</notifyURL>
    <callbackData>1234</callbackData>
  </callbackReference>
  <clientCorrelator>321</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <duration>7200</duration>
  <frequency>600</frequency>
</pr:presenceSubscription>
```

5.25.5.2.2 Response

```
HTTP/1.1 404 Not Found
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<common:requestError xmlns:common="urn:oma:xml:rest:common:1">
  <link rel="PresenceSourceList"
    href="http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/presenceSources"/>
  <serviceException>
    <messageId>SVC0004</messageId>
    <text>No valid addresses provided in message part %1</text>
    <variables>The specified identity does not exists.</variables>
  </serviceException>
</common:requestError>
```

5.25.6 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 section 14.7 of [RFC 2616].

5.26 Resource: Individual Presence subscription

The resource used is:

http://{serverRoot}/{apiVersion}/presence/{userId}/subscriptions/presenceSubscriptions/{presentityUserId}/{subscriptionId}

The userId, presentityUserId and subscriptionId must be percent-encoded according to [RFC3986].

This resource is used by a Watcher to manage an individual Presence subscription.

5.26.1 Request URI variables

The following request URI variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com/exampleAPI
apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)
userId	identity of the Watcher managing the subscription. Example: tel:+1-555-101
presentityUserId	identity of the Presentity owning the presence information Example: tel:+1-555-100
subscriptionId	identifier of the subscription generated by the system

5.26.2 Response Codes

5.26.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.26.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Presence, see Section 6.

5.26.3 GET

This operation is used by a Watcher to retrieve an individual Presence subscription.

5.26.3.1 Example: retrieving individual Presence subscription (Informative)

5.26.3.1.1 Request

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceSubscriptions/tel%3A%2B1-555-100/sub001 HTTP/1.1
Host: example.com:80
Accept: application/xml
```

5.26.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

```
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:presenceSubscription xmlns:pr="urn:oma:xml:rest:presence:1">
  <presentityUserId>tel:+1-555-100</presentityUserId>
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/presenceNotification</notifyURL>
    <callbackData>1234</callbackData>
  </callbackReference>
  <clientCorrelator>321</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <duration>5246</duration>
  <frequency>600</frequency>
  <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceSubscriptions/tel%3A%2B1-
555-100/sub001
  </resourceURL>
</pr:presenceSubscription>
```

5.26.4 PUT

This operation is used by a Watcher to update subscription parameters (e.g. duration, filter, frequency etc) for an ongoing Presence subscription.

5.26.4.1 Example: updating individual Presence subscription (Informative)

5.26.4.1.1 Request

```
PUT /exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceSubscriptions/tel%3A%2B1-555-100/sub001 HTTP/1.1
Host: example.com:80
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:presenceSubscription xmlns:pr="urn:oma:xml:rest:presence:1">
  <presentityUserId>tel:+1-555-100</presentityUserId>
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/presenceNotification</notifyURL>
    <callbackData>1234</callbackData>
  </callbackReference>
  <clientCorrelator>321</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <duration>7200</duration>
  <frequency>600</frequency>
</pr:presenceSubscription>
```

5.26.4.1.2 Response

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Length: nnnn
```

```
<?xml version="1.0" encoding="UTF-8"?>
<pr:presenceSubscription xmlns:pr="urn:oma:xml:rest:presence:1">
  <presentityUserId>tel:+1-555-100</presentityUserId>
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/presenceNotification</notifyURL>
    <callbackData>1234</callbackData>
  </callbackReference>
  <clientCorrelator>321</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <duration>7200</duration>
  <frequency>600</frequency>
  <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceSubscriptions/tel%3A%2B1-
555-100/sub001
  </resourceURL>
</pr:presenceSubscription>
```

5.26.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, PUT, DELETE' field in the response as per section 14.7 of [RFC 2616].

5.26.6 DELETE

This operation is used by a Watcher to terminate an active Presence subscription.

5.26.6.1 Example: terminating individual Presence subscription (Informative)

5.26.6.1.1 Request

```
DELETE /exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceSubscriptions/tel%3A%2B1-555-100/sub001HTTP/1.1
Host: example.com:80
```

5.26.6.1.2 Response

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

5.27 Resource: Presence notification

The resource URL is provided by the Watcher client when the subscription was created. ParlayREST does not make any assumption about the structure of this URL.

A notification is generated by the system in the following occasions:

Type of notification	Generated in the following occasions:

Initial notification	The subscription was successfully created (authorization decision evaluated to 'Allow', 'Politely-block' or 'Confirm').
Subsequent notification	A change in the presence information by the Presentity. Please note that a request to extend the duration of a subscription does not generate a new notification.
Final notification	An 'Active' or 'Pending' subscription that was blocked by the Presentity An 'Active' or 'Pending' subscription where the Presentity was removed from the system. An 'Active' or 'Pending' subscription that was terminated for an unknown reason.

5.27.1 Request URI variables

Provided by the Watcher client

5.27.2 Response Codes

5.27.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.27.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Presence, see [3GPP 29.199-14].

5.27.3 GET

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, PUT, DELETE' field in the response as per section 14.7 of [RFC 2616].

5.27.4 PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, PUT, DELETE' field in the response as per section 14.7 of [RFC 2616].

5.27.5 POST

This operation is used by the system when a new notification is generated.

5.27.5.1 Example 1: notifying Watcher about Presence data updates from an active subscription (Informative)

5.27.5.1.1 Request

```
POST /notifications/presenceNotification HTTP/1.1
Host: example.com:80
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:presenceNotification xmlns:pr="urn:oma:xml:rest:presence:1">
  <presentityUserId>tel:+1-555-100</presentityUserId>
  <callbackData>1234</callbackData>
```

```
 <resourceStatus>Active</resourceStatus>
 <presence>
  <person>
   <mood>
    <moodValue>Happy</moodValue>
   </mood>
  </person>
 </presence>
 <link rel="PresenceSubscription"
  href="http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceSubscriptions/tel%3A%2B1-555-
100/sub001"/>
</pr:presenceNotification>
```

5.27.5.1.2 Response

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

5.27.5.2 Example 2: notifying Watcher about Presence data updates from pending subscription (Informative)

5.27.5.2.1 Request

```
POST /notifications/presenceNotification HTTP/1.1
Host: example.com:80
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:presenceNotification xmlns:pr="urn:oma:xml:rest:presence:1">
 <presentityUserId>tel:+1-555-100</presentityUserId>
 <callbackData>2345</callbackData>
 <resourceStatus>Pending</resourceStatus>
 <link rel="PresenceSubscription"
  href="http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceSubscriptions/tel%3A%2B1-555-
101/sub002"/>
</pr:presenceNotification>
```

5.27.5.2.2 Response

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

5.27.5.3 Example 3: notifying Watcher about termination of Presence subscription (reason unknown) (Informative)

5.27.5.3.1 Request

```
POST /notifications/presenceNotification HTTP/1.1
Host: example.com:80
```

```
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:presenceNotification xmlns:pr="urn:oma:xml:rest:presence:1">
  <presentityUserId>tel:+1-555-100</presentityUserId>
  <callbackData>1234</callbackData>
  <resourceStatus>TerminatedOther</resourceStatus>
 <link rel="PresenceSubscription"
  href="http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceSubscriptions/tel%3A%2B1-555-
100/sub001"/>
</pr:presenceNotification>
```

5.27.5.3.2 Response

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

5.27.5.4 Example 4: notifying Watcher about termination of Presence subscription (Watcher blocked) (Informative)

5.27.5.4.1 Request

```
POST /notifications/presenceNotification HTTP/1.1
Host: example.com:80
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:presenceNotification xmlns:pr="urn:oma:xml:rest:presence:1">
  <presentityUserId>tel:+1-555-100</presentityUserId>
  <callbackData>1234</callbackData>
  <resourceStatus>TerminatedBlocked</resourceStatus>
  <link rel="PresenceSubscription"
   href="http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceSubscriptions/tel%3A%2B1-555-
100/sub001"/>
</pr:presenceNotification>
```

5.27.5.4.2 Response

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

5.27.6 DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET, PUT, DELETE’ field in the response as per section 14.7 of [RFC 2616].

5.28 Resource: All presence list subscriptions

The resource used is:

http://{serverRoot}/{apiVersion}/presence/{userId}/subscriptions/presenceListSubscriptions

The userId must be percent-encoded according to [RFC3986].

This resource is used to retrieve all active Presence list subscriptions towards all Presence lists.

5.28.1 Request URI variables

The following request URI variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com/exampleAPI
apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)
userId	identity of the Watcher retrieving the subscriptions. Example: tel:+1-555-101

5.28.2 Response Codes

5.28.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.28.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Presence, see [3GPP 29.199-14].

5.28.3 GET

This operation is used by a Watcher to retrieve all active Presence list subscriptions towards all Presence lists.

5.28.3.1 Example: retrieving all Presence list subscriptions towards all Presence lists (Informative)

5.28.3.1.1 Request

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceListSubscriptions HTTP/1.1
Host: example.com:80
Accept: application/xml
```

5.28.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
```

```
<pr:presenceListSubscriptionCollection xmlns:pr="urn:oma:xml:rest:presence:1">
 <presenceListSubscription>
  <presenceListId>myFriends</presenceListId>
  <callbackReference>
   <notifyURL>http://application.example.com/notifications/presenceListNotification</notifyURL>
   <callbackData>1234</callbackData>
  </callbackReference>
  <clientCorrelator>321</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <duration>5246</duration>
  <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-
101/subscriptions/presenceListSubscriptions/myFriends/
   sub001</resourceURL>
 </presenceListSubscription>
 <presenceListSubscription>
  <presenceListId>myColleagues</presenceListId>
  <callbackReference>
   <notifyURL>http://application.example.com/notifications/presenceListNotification</notifyURL>
   <callbackData>4321</callbackData>
  </callbackReference>
  <clientCorrelator>123</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <duration>5237</duration>
  <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-
101/subscriptions/presenceListSubscriptions/myColleagues/
   sub002</resourceURL>
 </presenceListSubscription>
 <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-
101/subscriptions/presenceListSubscriptions</resourceURL>
</pr:presenceListSubscriptionCollection>
```

5.28.4 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 section 14.7 of [RFC 2616].

5.28.5 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 section 14.7 of [RFC 2616].

5.28.6 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 section 14.7 of [RFC 2616].

5.29 Resource: Presence list subscriptions

The resource used is:

http://{serverRoot}/{apiVersion}/presence/{userId}/subscriptions/presenceListSubscriptions/{presenceListId}

The userId and presenceListId must be percent-encoded according to [RFC3986].

This resource is used by a Watcher to manage Presence list subscriptions.

5.29.1 Request URI variables

The following request URI variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com/exampleAPI
apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)
userId	identity of the Watcher managing the subscriptions. Example: tel:+1-555-101
presenceListId	identity of the Presence list Example: myFriends

5.29.2 Response Codes

5.29.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.29.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Presence, see Section 6.

5.29.3 GET

This operation is used by a Watcher to retrieve all active Presence list subscriptions for the specified Presence list.

5.29.3.1 Example: retrieving all Presence list subscriptions towards a single Presence list (Informative)

5.29.3.1.1 Request

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceListSubscriptions/myFriends HTTP/1.1
Host: example.com:80
Accept: application/xml
```

5.29.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:presenceListSubscriptionCollection xmlns:pr="urn:oma:xml:rest:presence:1">
 <presenceListSubscription>
  <presenceListId>myFriends</presenceListId>
  <callbackReference>
   <notifyURL>http://application.example.com/notifications/presenceListNotification</notifyURL>
   <callbackData>1234</callbackData>
```

```
   </callbackReference>
   <clientCorrelator>321</clientCorrelator>
   <applicationTag>myApp</applicationTag>
   <duration>5246</duration>
   <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-
101/subscriptions/presenceListSubscriptions/myFriends/
    sub001</resourceURL>
  </presenceListSubscription>
  <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-
101/subscriptions/presenceListSubscriptions</resourceURL>
</pr:presenceListSubscriptionCollection>
```

5.29.4 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 section 14.7 of [RFC 2616].

5.29.5 POST

This operation is used by a Watcher to create a new Presence list subscription towards the specified Presence list.

5.29.5.1 Example: creating new Presence list subscription towards a single Presence list (Informative)

5.29.5.1.1 Request

```
POST /exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceListSubscriptions/myFriends HTTP/1.1
Host: example.com:80
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:presenceListSubscription xmlns:pr="urn:oma:xml:rest:presence:1">
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/presenceListNotification</notifyURL>
    <callbackData>1234</callbackData>
  </callbackReference>
  <clientCorrelator>321</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <duration>7200</duration>
  <presenceFilter>person/mood</presenceFilter>
  <presenceFilter>service/org.openmobilealliance:IM-Session/1.0</presenceFilter>
  <frequency>600</frequency>
</pr:presenceListSubscription>
```

5.29.5.1.2 Response

```
HTTP/1.1 201 Created
Location: http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceListSubscriptions/myFriends/sub001
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn
```

```
<?xml version="1.0" encoding="UTF-8"?>
<pr:presenceListSubscription xmlns:pr="urn:oma:xml:rest:presence:1">
  <presenceListId>myFriends</presenceListId>
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/presenceListNotification</notifyURL>
    <callbackData>1234</callbackData>
  </callbackReference>
  <clientCorrelator>321</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <duration>7200</duration>
  <presenceFilter>person/mood</presenceFilter>
  <presenceFilter>service/org.openmobilealliance:IM-Session/1.0</presenceFilter>
  <frequency>600</frequency>
  <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceListSubscriptions/myFriends/
   sub001</resourceURL>
</pr:presenceListSubscription>
```

5.29.6 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 section 14.7 of [RFC 2616].

5.30 Resource: Individual presence list subscription

The resource used is:

http://{serverRoot}/{apiVersion}/presence/{userId}/subscriptions/presenceListSubscriptions/{presenceListId}/{subscriptionId}

The userId, presenceListId and subscriptionId must be percent-encoded according to [RFC3986].

This resource is used by a Watcher to manage an individual Presence list subscription.

5.30.1 Request URI variables

The following request URI variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com/exampleAPI
apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)
userId	identity of the Watcher managing the subscription. Example: tel:+1-555-101
presenceListId	identity of the Presence list Example: myFriends
subscriptionId	identifier of the subscription generated by the system

5.30.2 Response Codes

5.30.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.30.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Presence, see Section 6.

5.30.3 GET

This operation is used by a Watcher to retrieve an individual Presence list subscription.

5.30.3.1 Example: retrieving individual Presence list subscription (Informative)

5.30.3.1.1 Request

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceListSubscriptions/myFriends/sub001 HTTP/1.1
Host: example.com:80
Accept: application/xml
```

5.30.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<pr:presenceListSubscription xmlns:pr="urn:oma:xml:rest:presence:1">
  <presenceListId>myFriends</presenceListId>
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/presenceListNotification</notifyURL>
    <callbackData>1234</callbackData>
  </callbackReference>
  <clientCorrelator>321</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <duration>5274</duration>
  <presenceFilter>person/mood</presenceFilter>
  <presenceFilter>service/org.openmobilealliance:IM-Session/1.0</presenceFilter>
  <frequency>600</frequency>
  <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceListSubscriptions/myFriends/
   sub001</resourceURL>
</pr:presenceListSubscription>
```

5.30.4 PUT

This operation is used by a Watcher to update subscription parameters (e.g. duration, filter, frequency etc) for an ongoing Presence list subscription.

5.30.4.1 Example: updating individual Presence list subscription (Informative)

5.30.4.1.1 Request

```
PUT /exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceListSubscriptions/myFriends/sub001 HTTP/1.1
Host: example.com:80
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:presenceListSubscription xmlns:pr="urn:oma:xml:rest:presence:1">
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/presenceListNotification</notifyURL>
    <callbackData>1234</callbackData>
  </callbackReference>
  <clientCorrelator>321</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <duration>7200</duration>
  <presenceFilter>person/mood</presenceFilter>
  <presenceFilter>service/org.openmobilealliance:IM-Session/1.0</presenceFilter>
  <frequency>600</frequency>
</pr:presenceListSubscription>
```

5.30.4.1.2 Response

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:presenceListSubscription xmlns:pr="urn:oma:xml:rest:presence:1">
  <presenceListId>myFriends</presenceListId>
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/presenceListNotification</notifyURL>
    <callbackData>1234</callbackData>
  </callbackReference>
  <clientCorrelator>321</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <duration>7200</duration>
  <presenceFilter>person/mood</presenceFilter>
  <presenceFilter>service/org.openmobilealliance:IM-Session/1.0</presenceFilter>
  <frequency>600</frequency>
  <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceListSubscriptions/myFriends/
   sub001</resourceURL>
</pr:presenceListSubscription>
```

5.30.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET, PUT, DELETE’ field in the response as per section 14.7 of [RFC 2616].

5.30.6 DELETE

This operation is used by a Watcher to terminate an active Presence list subscription.

5.30.6.1 Example: terminating individual Presence list subscription (Informative)

5.30.6.1.1 Request

```
DELETE /exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceListSubscriptions/myFriends/sub001 HTTP/1.1
Host: example.com:80
```

5.30.6.1.2 Response

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

5.31 Resource: Presence list notification

The resource URL is provided by the Watcher client when the subscription was created. ParlayREST does not make any assumption about the structure of this URL.

A notification is generated by the system in the following occasions:

Type of notification	Generated in the following occasions:
Initial notification	The subscription was successfully created.
Subsequent notification	A change in the presence information by the Presentity. Please note that a request to extend the duration of a subscription does not generate a new notification.
Final notification	An 'Active' subscription where the Presentity was removed from the system. An 'Active' subscription that was terminated for an unknown reason.

5.31.1 Request URI variables

Provided by the Watcher client

5.31.2 Response Codes

5.31.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.31.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Presence, see [3GPP 29.199-14].

5.31.3 GET

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, PUT, DELETE' field in the response as per section 14.7 of [RFC 2616].

5.31.4 PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, PUT, DELETE' field in the response as per section 14.7 of [RFC 2616].

5.31.5 POST

This operation is used by the system when a new notification is generated.

5.31.5.1 Example 1: notifying Watcher about Presence data updates relating to Presence list (Informative)

5.31.5.1.1 Request

```
POST /notifications/presenceListNotification HTTP/1.1
Host: example.com:80
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:presenceListNotification xmlns:pr="urn:oma:xml:rest:presence:1">
 <presenceListId>myFriends</presenceListId>
 <callbackData>1234</callbackData>
 <resourceStatus>Active</resourceStatus>
 <presenceList>
  <presenceContact>
   <presentityUserId>tel:+1-555-101</presentityUserId>
   <resourceStatus>Active</resourceStatus>
   <presence>
    <person>
     <mood>
      <moodValue>Happy</moodValue>
     </mood>
    </person>
   </presence>
  </presenceContact>
  <presenceContact>
   <presentityUserId> tel:+1-555-102</presentityUserId>
   <resourceStatus>Pending</resourceStatus>
  </presenceContact>
 </presenceList>
 <link rel="PresenceListSubscription"
  href="http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-
100/subscriptions/presenceListSubscriptions/myFriends/sub001"/>
</pr:presenceListNotification>
```

5.31.5.1.2 Response

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

5.31.5.2 Example 2: notifying Watcher about termination of Presence list subscription (No resource) (Informative)

5.31.5.2.1 Request

```
POST /notifications/presenceListNotification HTTP/1.1
Host: example.com:80
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:presenceListNotification xmlns:pr="urn:oma:xml:rest:presence:1">
  <presenceListId>myFriends</presenceListId>
  <callbackData>1234</callbackData>
  <resourceStatus>TerminatedNoResource</resourceStatus>
</pr:presenceListNotification>
```

5.31.5.2.2 Response

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

5.31.5.3 Example 3: notifying Watcher about termination of Presence subscription (reason unknown) (Informative)

5.31.5.3.1 Request

```
POST /notifications/presenceListNotification HTTP/1.1
Host: example.com:80
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:presenceListNotification xmlns:pr="urn:oma:xml:rest:presence:1">
  <presenceListId>myFriends</presenceListId>
  <callbackData>1234</callbackData>
  <resourceStatus>TerminatedOther</resourceStatus>
  <link rel="PresenceListSubscription"
   href="http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-
101/subscriptions/presenceListSubscriptions/myFriends/sub001"/>
</pr:presenceListNotification>
```

5.31.5.3.2 Response

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

5.31.5.4 Example 4: notifying Watcher about subscription time out (Informative)

5.31.5.4.1 Request

```
POST /notifications/presenceListNotification HTTP/1.1
Host: example.com:80
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:presenceListNotification xmlns:pr="urn:oma:xml:rest:presence:1">
  <presenceListId>myFriends</presenceListId>
  <callbackData>1234</callbackData>
  <resourceStatus>TerminatedTimeout</resourceStatus>
  <link rel="PresenceListSubscription"
   href="http://example.com/exampleAPI/1/presence/{userId}/subscriptions/presenceListSubscriptions/myFriends/sub001"/>
</pr:presenceListNotification>
```

5.31.5.4.2 Response

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

5.31.6 DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, PUT, DELETE' field in the response as per section 14.7 of [RFC 2616].

6. Fault definitions

6.1 Service Exceptions

The following Service Exception codes are defined for the ParlayREST Presence in addition to those in [3GPP 29.199-14].

6.1.1 SVC0222: Key property changes not allowed

Element name	Description
MessageId	SVC0222
Text	Key property changes not allowed: key property %1
Variables	%1 – key property

6.2 Policy Exceptions

The following Policy Exception codes are defined for the ParlayREST Presence in addition to those in [3GPP 29.199-14].

6.2.1 POL0260: Too many resources

Element name	Description
MessageId	POL0260
Text	Maximum number of permitted resources exceeded.
Variables	None

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
n/a	n/a	No prior version –or- No previous version within OMA

A.2 Draft/Candidate Version 1.0 History

Document Identifier	Date	Sections	Description
Draft Versions: OMA-TS-ParlayREST_Presence-V1_0	15 Apr 2010	Many	TS skeleton prepared
	26 May 2010	5.1, 5.2, 5.3	CRs implemented: OMA-ARC-REST-2010-0225R03-CR_Presence_resource_summary OMA-ARC-REST-2010-0226R03-CR_Presence_data_structures OMA-ARC-REST-2010-0227R03-CR_Presence_sequence_diagrams
	09 Jul 2010	Many	CRs implemented: OMA-ARC-REST-2010-0271R02-CR_Presence_resources_Xian_comments_resolution OMA-ARC-REST-2010-0272R03-CR_Presence_data_structures_Xian_comments_resolution OMA-ARC-REST-2010-0288R01-CR_Presence_sequence_diagrams_updates OMA-ARC-REST-2010-0289R01-CR_Presence_TS_updates_sections_2to5 OMA-ARC-REST-2010-0290R03-CR_Presence_TS_resource_operations_XML_examples Editorial changes: Resource tables sorted to align with resource tree in section 5.1. Sections 5.4 -5.32 sorted also to align with the resource tables in section 5.1. Data structures sorted to align with resource tables and to move enumerations at the end of the data structures. Presence XML schema based on this version of the TS needs to be created
	10 Sep 2010	Many	CRs implemented: OMA-ARC-REST-2010-390-CR_SCRs_for_Presence, OMA-ARC-REST-2010-391R01-CR_JSON_examples_for_Presence, OMA-ARC-REST-2010-392R03-CR_FormUrlEncoded_examples_for_Presence, OMA-ARC-REST-2010-393R01-CR_More_XML_examples_for_Presence, OMA-ARC-REST-2010-396R03-CR_Add_WatchersNotification_Example_To_Presence_TS, OMA-ARC-REST-2010-397R03-CR_Add_PresenceNotification_Example_To_Presence_TS, OMA-ARC-REST-2010-398R03-CR_Add_PresenceListNotification_Example_To_Presence_TS, OMA-ARC-REST-2010-0417-CR_Presence_TS_cleanup, OMA-ARC-REST-2010-0436-CR_Presence_XML_example_subheadings, OMA-ARC-REST-2010-0466-CR_Delete_Duration_Element_from_Notification_Examples_in_presence_TS OMA-ARC-REST-2010-0474-CR_Presence_resolution_AI_68_and_70, OMA-ARC-REST-2010-0476-CR_Presence_resolution_AI_63, OMA-ARC-REST-2010-0477-CR_Presence_resolution_AI_83, OMA-ARC-REST-2010-0490R01-CR_Presence_TS_editor_notes_resolution, XML examples corrected after validation. JSON examples added. XSD schema needs to be updated.
	30 Sep 2010	Many	CRs implemented: OMA-ARC-REST-2010-0517R02-CR_Presence_TS_small_corrections, OMA-ARC-REST-2010-0516R01-CR_Presence_update_references_to_Common_TSs XSD schema needs to be updated.
	07 Oct 2010	All	Editorial fixes: styles as per template
	26 Nov 2010	Many	CRs implemented:

Document Identifier	Date	Sections	Description
			OMA-ARC-REST-2010-0597-CR_CONRR_LanguageString_TS_Presence, OMA-ARC-REST-2010-0630-CR_ A150_CONRR_Comment_J015_Presence OMA-ARC-REST-2010-0631R01-CR_ A153_CONRR_Comment_J023_Presence, OMA-ARC-REST-2010-0639R01-CR_ A147_CONRR_Comment_J023_Presence, OMA-ARC-REST-2010-0640R01-CR_ A146_CONRR_Comment_J003_Presence, OMA-ARC-REST-2010-0641R01-CR_ A149_CONRR_Comment_J012_Presence, OMA-ARC-REST-2010-0642R01-CR_ A151_CONRR_Comment_J016_Presence, OMA-ARC-REST-2010-0646-CR_ A155_A156_CONRR_comments_J026_J034_J037a_Presence, OMA-ARC-REST-2010-0660R01-CR_ Comment_towards__network_element_of_Presence_ParlayREST_Presence_ TS, OMA-ARC-REST-2010-0661-CR_ A129_resFormat_in_Presence, OMA-ARC-REST-2010-0662R01-CR_ PlaceType_modification_in_Presence_TS OMA-ARC-REST-2010-0665-CR_ Presence_Sphere_Activity_Mood_changes AI implemeneted: - REST-2010-A122, A124, A125, A126, A127, A128, A129, A136, A138, A139, A152, A195
	12 Dec 2010	Many	CRs implemented: OMA-ARC-REST-2010- 0656-CR_Resolution_to_J004, OMA-ARC-REST-2010- 0657R01-CR_Resolution_to_J026, OMA-ARC-REST-2010- 0688- CR_A222_A223_CONRR_comments_J029_J031_for_Presence, OMA-ARC-REST-2010- 0689R01-CR_A236_CONRR_J001_for_Presence, OMA-ARC-REST-2010- 0691R01- CR_A239_CONRR_comment_J014_for_Presence, OMA-ARC-REST-2010- 0692R01- CR_A221_CONRR_comment_J022_for_Presence, OMA-ARC-REST-2010- 0699R01- CR_A157_224_245_CONRR_comments_J041_42_43a_for_Presence, OMA-ARC-REST-2010- 0701- CR_A238_CONRR_comment_J011_for_Presence, OMA-ARC-REST-2010- 0702- CR_A211_CONRR_G047_resolution_for_Presence, OMA-ARC-REST-2010- 0703R01- CR_CONRR_comment_A011_resolution_for_Presence, OMA-ARC-REST-2010- 0706- CR_A188_CONRR_comment_D005_resolution_for_Presence, OMA-ARC-REST-2010- 0707- CR_A220_CONRR_comment_J010_for_Presence OMA-ARC-REST-2010- 0709-CR_LinkList_extension_for_Presence, OMA-ARC-REST-2010- 0725- CR_Clarifications_of_authorization_rules_for_Presence. AI implemeneted: REST-2010-A278, A295, A303, A305, A309
	13 Dec 2010	All	Editorial fixes: heading styles, Contents page
	14 Dec 2010	C.2.1.1, C.3.1.1, C.4.1.1	In the request examples in these sections, the string ":+" that forms a part of telephone number has been replaced with the string "%3A%2B", according to the rules for percent-encoding of URI reserved characters.
	30 Dec 2010	Many	Implemented CR, OMA-ARC-REST-2010- 0735- CR_Fixing_XML_errors_from_validation_for_TS_Presence
Candidate Version: OMA-TS-ParlayREST_Presence-V1_0	11 Jan 2011	All	Status changed to Candidate by TP: OMA-TP-2010-0531R01-INP_ParlayREST_2_0_for_Candidate_approval

Appendix B. Static Conformance Requirements (Normative)

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

B.1 SCR for ParlayREST.Presence Server

Item	Function	Reference	Requirement
PARLAYREST-PRESENCE-SUPPORT-S-001-M	Support for the Presence REST API	5	
PARLAYREST-PRESENCE-SUPPORT-S-002-M	Support for the XML request & response format	5	
PARLAYREST-PRESENCE-SUPPORT-S-003-M	Support for the JSON request & response format	5	
PARLAYREST-PRESENCE-SUPPORT-S-004-O	Support for the application/form-urlencoded format	Appendix C	

B.1.1 SCR for ParlayREST.Presence.Presentity.PresenceSource Server

Item	Function	Reference	Requirement
PARLAYREST-PRESENCE-PRES-PS-S-001-M	Support for creation and retrieval of presence source data	5.4	
PARLAYREST-PRESENCE-PRES-PS-S-002-M	Retrieve all presence sources related to Presentity - GET	5.4.3	
PARLAYREST-PRESENCE-PS-S-003-M	Create presence source data - POST	5.4.5	

B.1.2 SCR for ParlayREST.Presence.Presentity.Individual.PresenceSource Server

Item	Function	Reference	Requirement
PARLAYREST-PRESENCE-PRES-IND-PS-S-001-M	Support for the management of presence data for a Presentity from an individual presence source	5.5	
PARLAYREST-PRESENCE-PRES-IND-PS-S-002-M	Retrieve presence data for a Presentity - GET	5.5.3	
PARLAYREST-PRESENCE-PRES-IND-PS-S-003-M	Update presence data for a Presentity - PUT	5.5.4	
PARLAYREST-PRESENCE-PRES-IND-PS-S-004-M	Delete presence data for a Presentity- DELETE	5.5.6	

B.1.3 SCR for ParlayREST.Presence.Presentity.Individual.PresenceSource.Attribute Server

Item	Function	Reference	Requirement
PARLAYREST-PRESENCE-PRES-IND-PS-ATTR-S-001-O	Support for the management of an individual presence attribute	5.6	PARLAYREST-PRESENCE-PRES-IND-PS-ATTR-S-003-O PARLAYREST-PRESENCE-PRES-IND-PS-ATTR-S-004-O
PARLAYREST-PRESENCE-PRES-IND-PS-ATTR-S-002-O	Retrieve value of an individual presence attribute - GET	5.6.3	
PARLAYREST-PRESENCE-PRES-IND-PS-ATTR-S-003-O	Update value of an individual presence attribute - PUT	5.6.4	
PARLAYREST-PRESENCE-PRES-IND-PS-ATTR-S-004-O	Delete value of an individual presence attribute - DELETE	5.6.6	

B.1.4 SCR for ParlayREST.Presence.Presentity.PresenceSource.Persistent Server

Item	Function	Reference	Requirement
PARLAYREST-PRESENCE-PRES-PRSOURCE-PERS-S-001-M	Support for the management of persistent presence data for a Presentity	5.7	
PARLAYREST-PRESENCE-PRES-PS-PERS-S-002-M	Retrieve persistent presence data - GET	5.7.3	
PARLAYREST-PRESENCE-PRES-PS-PERS-S-003-M	Update persistent presence data - PUT	5.7.4	
PARLAYREST-PRESENCE-PRES-PS-PERS-S-004-M	Delete persistent presence data - DELETE	5.7.6	

B.1.5 SCR for ParlayREST.Presence.Presentity.PresenceSource.Persistent.Attribute Server

Item	Function	Reference	Requirement
PARLAYREST-PRESENCE-PRES-PS-PERS-ATTR-S-001-O	Support for the management of an individual persistent	5.8	PARLAYREST-PRESENCE-

Item	Function	Reference	Requirement
	presence attribute		PRES-PS-PERS-ATTR-S-003-O PARLAYREST-PRESENCE-PRES-PS-PERS-ATTR-S-004-O
PARLAYREST-PRESENCE-PRES-PS-PERS-ATTR-S-002-O	Retrieve individual persistent presence attribute - GET	5.8.3	
PARLAYREST-PRESENCE-PRES-PS-PERS-ATTR-S-003-O	Update individual persistent presence attribute - PUT	5.8.4	
PARLAYREST-PRESENCE-PRES-PS-PERS-ATTR-S-004-O	Delete individual persistent presence attribute - DELETE	5.8.6	

B.1.6 SCR for ParlayREST.Presence.Presentity.ContentList Server

Item	Function	Reference	Requirement
PARLAYREST-PRESENCE-PRES-CONTL-S-001-M	Support for a read access to a content list	5.9	
PARLAYREST-PRESENCE-PRES-CONTL-S-002-M	Retrieve a content list -GET	5.9.3	

B.1.7 SCR for ParlayREST.Presence.Presentity.Individual.Content Server

Item	Function	Reference	Requirement
PARLAYREST-PRESENCE-PRES-IND-CONT-S-001-M	Support for the management of an individual content by Presentity	5.10	
PARLAYREST-PRESENCE-PRES-IND-CONT-S-002-M	Retrieve individual content - GET	5.10.3	
PARLAYREST-PRESENCE-PRES-IND-CONT-S-003-M	Create/replace individual content - PUT	5.10.4	
PARLAYREST-PRESENCE-PRES-IND-CONT-S-004-M	Delete (remove) individual content - DELETE	5.10.6	

B.1.8 SCR for ParlayREST.Presence.Presentity.WatcherList Server

Item	Function	Reference	Requirement
PARLAYREST-PRESENCE-PRES-WL-S-001-M	Support for a read access to Watcher list	5.11	
PARLAYREST-PRESENCE-PRES-WL-S-002-M	Retrieve Watcher list - GET	5.11.3	

B.1.9 SCR for ParlayREST.Presence.Presentity.Individual.Watcher Server

Item	Function	Reference	Requirement
PARLAYREST-PRESENCE-PRES-IND-WATCHER-S-001-M	Support for a read access to an individual Watcher information	5.12	
PARLAYREST-PRESENCE-PRES-IND-WATCHER-S-002-M	Retrieve individual Watcher information - GET	5.12.3	

B.1.10 SCR for ParlayREST.Presence.Presentity.Authorisation.Rules Server

Item	Function	Reference	Requirement
PARLAYREST-PRESENCE-PRES-AUTH-RULES-S-001-M	Support for creation and read access for authorisation rules	5.13	
PARLAYREST-PRESENCE-PRES-AUTH-RULES-S-002-M	Retrieve list of authorisation - GET	5.13.3	
PARLAYREST-PRESENCE-PRES-AUTH-RULES-S-003-M	Create a new authorisation rule - POST	5.13.5	

B.1.11 SCR for ParlayREST.Presence.Presentity.Individual.Authorisation.Rule Server

Item	Function	Reference	Requirement
PARLAYREST-PRESENCE-PRES-IND-AUTH-RULE-S-001-M	Support for the management of an individual authorisation rule	5.14	
PARLAYREST-PRESENCE-PRES-IND-AUTH-RULE-S-002-M	Retrieve an individual authorisation rule - GET	5.14.3	
PARLAYREST-PRESENCE-PRES-IND-AUTH-RULE-S-003-M	Update an individual authorisation rule - PUT	5.14.4	
PARLAYREST-PRESENCE-PRES-IND-AUTH-RULE-S-004-M	Delete an individual authorisation rule - DELETE	5.14.6	

B.1.12 SCR for ParlayREST.Presence.Presentity.Individual.Authorisation.Rule.Data Server

Item	Function	Reference	Requirement
PARLAYREST-PRESENCE-PRES-IND-AUTH-RULE-DATA-S-001-O	Support for the management of individual data from an authorisation rule	5.15	PARLAYREST-PRESENCE-PRES-IND-

Item	Function	Reference	Requirement
			AUTH-RULE-DATA-S-003-O PARLAYREST-PRESENCE-PRES-IND-AUTH-RULE-DATA -S-004-O
PARLAYREST-PRESENCE-PRES-IND-AUTH-RULE-DATA-S-002-O	Retrieve individual data from an authorisation rule - GET	5.15.3	
PARLAYREST-PRESENCE-PRES-IND-AUTH-RULE-DATA-S-003-O	Create/Update individual data for an authorisation rule - PUT	5.15.4	
PARLAYREST-PRESENCE-PRES-IND-AUTH-RULE-DATA -S-004-O	Delete individual data for an authorisation rule - DELETE	5.15.6	

B.1.13 SCR for ParlayREST.Presence.Watcher.PresenceContact Server

Item	Function	Reference	Requirement
PARLAYREST-PRESENCE-WATCH-PC-S-001-M	Support for a read access for a composite presence data for a single Presence contact (Presentity)	5.16	
PARLAYREST-PRESENCE-WATCH-PC-S-002-M	Retrieve composite presence data for a Presentity - GET	5.16.3	

B.1.14 SCR for ParlayREST.Presence.Watcher.Individual.PresenceContact.Attribute Server

Item	Function	Reference	Requirement
PARLAYREST-PRESENCE-WATCH-IND-PC-ATTR-S-001-O	Support for a read access for an individual presence attribute for a Presentity	5.17	PARLAYREST-PRESENCE-WATCH-IND-PC-ATTR-S-002-O
PARLAYREST-PRESENCE-WATCH-IND-PC-ATTR-S-002-O	Retrieve individual presence attribute for a Presentity - GET	5.17.3	

B.1.15 SCR for ParlayREST.Presence.Watcher.PresenceList Server

Item	Function	Reference	Requirement
PARLAYREST-PRESENCE-WATCH-PL-S-001-M	Support for a read access to presence data for Presence list	5.18	
PARLAYREST-PRESENCE-WATCH-PL-S-002-M	Retrieve presence data for all users (Presentities) in a Presence list - GET	5.18.3	

B.1.16 SCR for ParlayREST.Presence.Watcher.PresenceContactContent Server

Item	Function	Reference	Requirement
PARLAYREST-PRESENCE-WATCH-PCC-S-001-M	Support for a read access for a content	5.19	
PARLAYREST-PRESENCE-WATCH-PCC-S-002-M	Retrieve a content from a Presentity - GET	5.19.3	

B.1.17 SCR for ParlayREST.Presence.Subscriptions Server

Item	Function	Reference	Requirement
PARLAYREST-PRESENCE-SUBSCR-S-001-O	Support for a read access for subscriptions for a particular user (could be either in Presentity or Watcher role)	5.20	PARLAYREST-PRESENCE-SUBSCR-S-002-O
PARLAYREST-PRESENCE-SUBSCR-S-002-O	Read all active subscriptions - GET	5.20.3	

B.1.18 SCR for ParlayREST.Presence.Presentity.Subscriptions.WatchersSubscriptions Server

Item	Function	Reference	Requirement
PARLAYREST-PRESENCE-PRES-SUBSCR-WS-S-001-O	Support for subscriptions for notifications about Watchers information	5.21	PARLAYREST-PRESENCE-PRES-SUBSCR-WS-S-003-O
PARLAYREST-PRESENCE-PRES-SUBSCR-WS-S-002-O	Read all active subscriptions - GET	5.21.3	
PARLAYREST-PRESENCE-PRES-SUBSCR-WS-S-003-O	Create subscription for a Watchers list - POST	5.21.5	

B.1.19 SCR for ParlayREST.Presence.Presentity.Individual.Subscriptions.WatchersSubscriptions Server

Item	Function	Reference	Requirement
PARLAYREST-PRESENCE-PRES-IND-SUBSCR-WS-S-001-O	Support for individual subscription for notifications about Watchers information	5.22	PARLAYREST-PRESENCE-PRES-IND-SUBSCR-WS-S-003-O PARLAYREST-PRESENCE-PRES-IND-SUBSCR-WS-S-004-O
PARLAYREST-PRESENCE-PRES-IND-SUBSCR-WS-S-002-O	Read individual subscription - GET	5.22.3	
PARLAYREST-PRESENCE-PRES-IND-SUBSCR-WS-S-003-O	Update (and/or extend duration of) subscription - PUT	5.22.4	
PARLAYREST-PRESENCE-PRES-IND-SUBSCR-WS-S-004-O	Delete (terminate) subscription - DELETE	5.22.6	

B.1.20 SCR for ParlayREST.Presence.WatchersSubscriptions.Notifications Server

Item	Function	Reference	Requirement
PARLAYREST-PRESENCE-WS-NOTIF-S-001-O	Support for notifying application (Presentity) about changes in Watcher's subscription status	5.23	PARLAYREST-PRESENCE-PRES-WS-S-002-O
PARLAYREST-PRESENCE-PRES-WS-S-002-O	Notify application about changes in Watcher's subscription status - POST	5.23.5	

B.1.21 SCR for ParlayREST.Presence.Watcher.Subscriptions.PresenceSubscriptions Server

Item	Function	Reference	Requirement
PARLAYREST-PRESENCE-WATCH-SUBSCR-PS-S-001-O	Support for read access for presence subscriptions	5.24	PARLAYREST-PRESENCE-WATCH-SUBSCR-PS-S-002-O
PARLAYREST-PRESENCE-WATCH-SUBSCR-PS-S-002-O	Read all active presence subscriptions - GET	5.24.3	

B.1.22 SCR for ParlayREST.Presence.Watcher.Subscriptions.PresenceSubscriptions.SinglePresentity Server

Item	Function	Reference	Requirement
PARLAYREST-PRESENCE-WATCH-SUBSCR-PS-SINGP-S-001-O	Support for subscriptions for notifications about presence information for a particular Presentity	5.25	PARLAYREST-PRESENCE-WATCH-SUBSCR-PS-SINGP-S-003-O
PARLAYREST-PRESENCE-WATCH-SUBSCR-PS-SINGP-S-002-O	Read all active presence subscriptions for a particular Presentity - GET	5.25.3	
PARLAYREST-PRESENCE-WATCH-SUBSCR-PS-SINGP-S-003-O	Create subscription for presence information for a particular Presentity - POST	5.25.5	

B.1.23 SCR for ParlayREST.Presence.Watcher.Individual.Subscriptions.PresenceSubscriptions.SinglePresentity Server

Item	Function	Reference	Requirement
PARLAYREST-PRESENCE-WATCH-IND-SUBSCR-PS-SINGP-S-001-O	Support for individual subscription for notifications about presence information for a particular Presentity	5.26	PARLAYREST-PRESENCE-WATCH-IND-SUBSCR-PS-SINGP-S-003-O PARLAYREST-PRESENCE-WATCH-IND-SUBSCR-PS-SINGP-S-004-O
PARLAYREST-PRESENCE-WATCH-IND-SUBSCR-PS-SINGP-S-002-O	Read an individual presence subscription for a particular Presentity - GET	5.26.3	
PARLAYREST-PRESENCE-WATCH-IND-SUBSCR-PS-SINGP-S-003-O	Update (and/or extend duration of) subscription for presence information for a particular Presentity - PUT	5.26.4	
PARLAYREST-PRESENCE-WATCH-IND-SUBSCR-PS-SINGP-S-004-O	Delete (terminate) subscription for	5.26.6	

Item	Function	Reference	Requirement
	presence information for a particular Presentity - DELETE		

B.1.24 SCR for ParlayREST.Presence.PresenceSubscriptions.Notifications Server

Item	Function	Reference	Requirement
PARLAYREST-PRESENCE-PS-NOTIF-S-001-O	Support for notifying application (Watcher) about changes in presence status	5.27	PARLAYREST-PRESENCE-PS-NOTIF-S-002-O
PARLAYREST-PRESENCE-PS-NOTIF-S-002-O	Notify application about changes in presence status - POST	5.27.5	

B.1.25 SCR for ParlayREST.Presence.Watcher.Subscriptions.PresenceListSubscriptions Server

Item	Function	Reference	Requirement
PARLAYREST-PRESENCE-WATCH-SUBSCR-PLS-S-001-O	Support for read access for all presence subscriptions for all Presence lists	5.28	PARLAYREST-PRESENCE-WATCH-SUBSCR-PLS-S-002-O
PARLAYREST-PRESENCE-WATCH-SUBSCR-PLS-S-002-O	Read all active presence list subscriptions towards all Presence lists - GET	5.28.3	

B.1.26 SCR for ParlayREST.Presence.Watcher.Subscriptions.PresenceListSubscriptions.SinglePresenceList Server

Item	Function	Reference	Requirement
PARLAYREST-PRESENCE-WATCH-SUBSCR-PLS-SINGPL-S-001-O	Support for presence list subscriptions for a single Presencelist	5.29	PARLAYREST-PRESENCE-WATCH-SUBSCR-PLS-SINGPL-S-003-O
PARLAYREST-PRESENCE-WATCH-SUBSCR-PLS-SINGPL-S-002-O	Read all active presence list subscriptions for a	5.29.3	

Item	Function	Reference	Requirement
	particular Presence list - GET		
PARLAYREST-PRESENCE-WATCH-SUBSCR-PLS-SINGPL-S-003-O	Create presence list subscription for a particular Presence list - POST	5.29.5	

B.1.27 SCR for ParlayREST.Presence.Watcher.Individual.Subscriptions.PresenceListSubscriptions.SinglePresenceList Server

Item	Function	Reference	Requirement
PARLAYREST-PRESENCE-WATCH-IND-SUBSCR-PLS-SINGPL-S-001-O	Support for individual presence list subscription for a single Presence list	5.30	PARLAYREST-PRESENCE-WATCH-IND-SUBSCR-PLS-SINGPL-S-003-O PARLAYREST-PRESENCE-WATCH-IND-SUBSCR-PLS-SINGPL-S-004-O
PARLAYREST-PRESENCE-WATCH-IND-SUBSCR-PLS-SINGPL-S-002-O	Read an individual presence list subscription for a particular Presence list - GET	5.30.3	
PARLAYREST-PRESENCE-WATCH-IND-SUBSCR-PLS-SINGPL-S-003-O	Update (and/or extend duration of) presence list subscription for a particular Presence list - PUT	5.30.4	
PARLAYREST-PRESENCE-WATCH-IND-SUBSCR-PLS-SINGPL-S-004-O	Delete (terminate) presence list subscription for a particular Presence list - DELETE	5.30.6	

B.1.28 SCR for ParlayREST.Presence.PresenceListSubscriptions.Notifications Server

Item	Function	Reference	Requirement
PARLAYREST-PRESENCE-PLS-NOTIF-S-001-O	Support for notifying application (Watcher) about changes in presence status for a	5.31	PARLAYREST-PRESENCE-PLS-NOTIF-S-

Item	Function	Reference	Requirement
	Presentity from a particular Presence list		002-O
PARLAYREST-PRESENCE-PLS-NOTIF-S-002-O	Notify application about changes in presence status for a Presentity from a particular Presence list - POST	5.31.5	

Appendix C. Application/x-www-form-urlencoded Request Format for POST Operations (Normative)

This section defines a format for the Presence REST API requests where the body of the request is encoded using the application/x-www-form-urlencoded MIME type.

Note: only the request body is encoded as application/x-www-form-urlencoded, the response is still encoded as XML or JSON depending on the preference of the client and the capabilities of the server.

The encoding is defined below for all Presence REST operations which are based on POST requests:

C.1 Create presence source

This operation is used for creating a presence source with a specified time-to-live. Creation of presence data using a form-urlencoded request is supported for "person" related presence data only. This specification does not support creation of presence sources that include "service" or "device" related presence data in form-urlencoded requests.

The request parameters for creation of presence source are as follows:

Name	Type/Values	Optional	Description
clientCorrelator	xsd:string	Yes	A correlator that the client MAY use to tag this particular resource representation during a request to create a resource on the server. In case the field is present, the server SHALL not alter its value, and SHALL provide it as part of the representation of this resource. In case the field is not present, the server SHALL NOT generate it.
applicationTag	xsd:string	Yes	A tag that the client MAY use to tag this particular resource on the server. In case the field is present, the server SHALL not alter its value, and SHALL provide it as part of the representation of this resource. In case the field is not present, the server SHALL NOT generate it.
duration	xsd:int	Yes	Specifies the life time in seconds. When this time has elapsed the publication will expire unless it has been refreshed (by using some other methods than form-urlencoded). If the parameter is omitted, the publication life time will be specified by the service policy. A too low value (including "0") will result in an error response. What is too low is defined by service policy. A too high requested value may be reduced by the server according to service policy.
person-activities	ActivityValue [0..unbounded]	Yes	The Presentity's activity (available, busy, lunch, etc.)
person-activities-note	xsd:string	Yes	A textual description of what the user is currently doing. MAY only be included if person-activities is specified.
person-activities-other	xsd:string	Yes	MAY only be included if person-activities is specified.
person-activities-until	xsd:dateTime	Yes	Indicates an absolute time the attribute is expected to

			be valid. MAY only be included if person-activities is specified.
person-placeType	PlaceTypeValue [0..unbounded]	Yes	Specifies the place type values.
person-placeType-note	xsd:string	Yes	A textual description of what type of place the person is located in. MAY only be included if person-placeType is specified.
person-placeType-other	xsd:string	Yes	MAY only be included if person-placeType is specified.
person-placeType-until	xsd:dateTime	Yes	Indicates an absolute time the attribute is expected to be valid. MAY only be included if person-placeType is specified.
person-privacy	PrivacyValue [0..unbounded]	Yes	Specifies the type of privacy expected at the user's current position.
person-privacy-note	xsd:string	Yes	A textual description of the privacy. MAY only be included if person-privacy is specified.
person-sphere	SphereValue	Yes	Specifies the sphere the user is in.
person-sphere-other	xsd:string	Yes	A textual description of the sphere. MAY only be included if person-sphere is specified.
person-mood	MoodValue [0..unbounded]	Yes	Specifies the mood of the user (happy, sad etc)
person-mood-note	xsd:string	Yes	A textual description of the mood. MAY only be included if person-mood is specified.
person-mood-other	xsd:string	Yes	MAY only be included if person-mood is specified.
person-mood-until	xsd:dateTime	Yes	Indicates an absolute time the attribute is expected to be valid. MAY only be included if person-mood is specified.
person-placeIs-Audio	PlaceIsAudio	Yes	Describes place conditions for audio communication.
person-placeIs-Video	PlaceIsVideo	Yes	Describes place conditions for video communication.
person-placeIs-Text	PlaceIsText	Yes	Describes place conditions for real-time and instant-messaging communication.
person-timeOffset	xsd:int	Yes	Number of minutes of offset from UTC that the user is currently at.
person-timeOffset-until	xsd:dateTime	Yes	Indicates an absolute time the attribute is expected to be valid. MAY only be included if person-timeOffset is specified.
person-statusIcon	xsd:anyURI	Yes	The URL of the content.
person-statusIcon-contentType	xsd:string	Yes	The content-type related to the content. MAY only be included if person-statusIcon is specified.
person-statusIcon-eTag	xsd:string	Yes	The ETag of the content. MAY only be included if

			person-statusIcon is specified.
person-statusIcon-fSize	xsd:int	Yes	The size of the content. MAY only be included if person-statusIcon is specified.
person-statusIcon-resolution	xsd:string	Yes	The resolution of the content. MAY only be included if person-statusIcon is defined.
person-statusIcon-until	xsd:dateTime	Yes	Indicates an absolute time the attribute is expected to be valid. MAY only be included if person-statusIcon is defined.
person-class	xsd:token	Yes	Specifies a class of the presence data.
person-note	xsd:string	Yes	Contains a tagline.
person-note-lang	xsd:string	Yes	Language of the text for a note. The format of this parameter is aligned with that of the built-in XML attribute xml:lang [W3C-XML11]. It is RECOMMENDED to provide this parameter
person-location-circle-latitude	xsd:float	Yes	Latitude of center point. MUST be included if person-location-circle-longitude is included.
person-location-circle-longitude	xsd:float	Yes	Longitude of center point. MUST be included if person-location-circle-latitude is included.
person-location-circle-radius	xsd:float	Yes	Radius of circle around center point in meters. MAY only be included if person-location-circle-latitude and person-location-circle-longitude is specified.
person-overridingWillingness	OpenOrClosed	Yes	The overriding willingness for a person.
person-overridingWillingness-until	xsd:dateTime	Yes	Specifies validity for the attribute. MAY only be included if person-overridingWillingness is specified.
person-link	xsd:anyURI	Yes	The address for the link
person-link-label	xsd:string	Yes	Label for the link. MAY only be included if person-link is specified.
person-link-priority	xsd:decimal	Yes	Priority for the link. MAY only be included if person-link is specified.
person-card	xsd:anyURI	Yes	URI to a business card.
person-displayName	xsd:string	Yes	A display name of a person.
person-homePage	xsd:anyURI	Yes	URI pointing to general information about a person.
person-icon	xsd:anyURI	Yes	URI pointing to an image/icon of the person. Note: It is recommended to use the StatusIcon for sharing icons/avatars between users.
person-map	xsd:anyURI	Yes	URI pointing to a map related to the person

person-sound	xsd:anyURI	Yes	URI pointing to a sound related to the person.

If the operation was successful, it returns an HTTP Status of "201 Created".

C.1.1 Example: creating presence source for user (Informative)

C.1.1.1 Request

```
POST /exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources HTTP/1.1
Accept: application/xml
Host: example.com:80
Content-Type: application/x-www-form-urlencoded
Content-Length: nnnn

clientCorrelator=123&
applicationTag=myApp&
duration=3600&
person-activities=Busy&
person-activities-note=meeting%20until%20lunch&
person-placeType=Home&
person-placeType-note=At%20home!&
person-mood=happy
```

C.1.1.2 Response

```
HTTP/1.1 201 Created
Location: http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/prs123
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:presenceSource xmlns:pr="urn:oma:xml:rest:presence:1">
  <clientCorrelator>123</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <duration>3600</duration>
  <presence>
    <person>
      <activities>
        <activityValue>Busy</activityValue>
        <note>meeting until lunch</note>
      </activities>
      <placeType>
        <placeTypeValue>Home</placeTypeValue>
        <note>At home</note>
      </placeType>
      <mood>
        <moodValue>Happy</moodValue>
      </mood>
    </person>
  </presence>
  <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/prs123</resourceURL>
</pr:presenceSource>
```

C.2 Create authorization rule

This operation is used by a Presentity to create an authorization rule. The authorization rules control who will have access to Presentity's presence information. A Watcher may be authorized to all or a subset of the available presence attributes.

The request parameters are as follows:

Name	Type/Values	Optional	Description
ruleName	xsd:ID	No	A name associated with the rule
watcherUserId	xsd:anyURI [0..unbounded]	Choice	Contains a list of Watcher identities for which the rule will apply
memberListId	xsd:string [0..unbounded]	Choice	Contains a list of member list identities for which the rule will apply
domainName	xsd:string [0..unbounded]	Choice	Contains a list of domain names for which the rule will apply
anonymous	xsd:boolean	Choice	Specifies that the rule will apply for anonymous requests. XSD modeling sets this parameter value to "true".
otherUser	xsd:boolean	Choice	Specifies that the rule will apply for unknown users. It allows the client to specify a default behavior for unknown users. XSD modeling sets this parameter value to "true"
decision	DefaultDecisionValue	No	The authorization decision for the rule.
presenceFilter	xsd:anyURI [0..unbounded]	Yes	Contains filter indicating which presence attributes the Watchers are allowed to see. An empty filter means that the Watchers have access to all presence attributes.

If the operation was successful, it returns an HTTP Status of "201 Created".

C.2.1 Example: creating an authorization rule (Informative)

C.2.1.1 Request

```
POST /exampleAPI/1/presence/tel%3A%2B1-555-100/authorization/rules HTTP/1.1
Accept: application/xml
Host: example.com:80
Content-Type: application/x-www-form-urlencoded
Content-Length: nnnn

ruleName=allowList&
watcherUserId=tel%3A%2B1-555-102&
watcherUserId=tel%3A%2B1-555-104&
watcherUserId=tel%3A%2B1-555-105&
decision=Allow
```

C.2.1.2 Response

```
HTTP/1.1 201 Created
Location: http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/authorization/rules/rule003
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:rule ruleName="allowList" xmlns:pr="urn:oma:xml:rest:presence:1">
  <watcherUserId>tel:+1-555-102</watcherUserId>
  <watcherUserId>tel:+1-555-104</watcherUserId>
  <watcherUserId>tel:+1-555-105</watcherUserId>
  <decision>Allow</decision>
<resourceURL> http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/authorization/rules/rule003</resourceURL>
</pr:rule>
```

C.3 Create Watchers subscription

This resource is used by a Presentity to create a Watchers subscription, i.e. subscription for changes in the Watchers list. The list contains Watchers that are subscribing for presence information about the Presentity.

The request parameters are as follows:

Name	Type/Value	Optional	Description
presentityUserId	xsd:anyURI	Yes	Identifies the Presentity for which the subscription is created towards. Mandatory in responses.
notifyURL	xsd:anyURI	No	Notification endpoint definition.
callbackData	xsd:string	Yes	Data the application can register with the server when subscribing to notifications, and that are passed back unchanged in each of the related notifications.
notificationFormat	common:NotificationFormat	Yes	Default: XML Application can specify format of the resource representation in notifications that are related to this subscription. The choice is between {XML, JSON}.
clientCorrelator	xsd:string	Yes	A correlator that the client MAY use to tag this particular resource representation during a request to create a resource on the server. In case the field is present, the server SHALL not alter its value, and SHALL provide it as part of the representation of this resource. In case the field is not present, the server SHALL NOT generate it.
applicationTag	xsd:string	Yes	A tag that the client MAY use to tag this particular resource on the server. In case the field is present, the server SHALL not alter its value, and SHALL provide it as part of the representation of this resource. In case the field is not present, the server SHALL NOT generate it.

duration	xsd:int	Yes	Specifies the duration of the subscription in seconds. When this time has elapsed the subscription will expire unless it has been refreshed (by using some other methods than form-urlencoded).
resourceStatusFilter	ResourceStatus [0..unbounded]	Yes	Indicates the desired Watcher subscription statuses that the Presentity is interested to get notifications about. If the parameter is omitted or there is an empty filter it means monitoring all states.
frequency	xsd:int	Yes	Maximum frequency of notifications, expressed as minimum time between notifications in seconds.

If the operation was successful, it returns an HTTP Status of "201 Created".

C.3.1 Example: creating new Watchers subscription (Informative)

C.3.1.1 Request

```
POST /exampleAPI/1/presence/tel%3A%2B1-555-100/subscriptions/watchersSubscriptions HTTP/1.1
Accept: application/xml
Host: example.com:80
Content-Type: application/x-www-form-urlencoded
Content-Length: nnnn

presentityUserId=tel%3A%2B1-555-100&
notifyURL=http://application.example.com/notifications/watchersNotification&
callbackData=1234&
clientCorrelator=321&
applicationTag=myApp&
duration=7200&
resourceStatusFilter=Pending
```

C.3.1.2 Response

```
HTTP/1.1 201 Created
Location: http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/subscriptions/watchersSubscriptions/sub001
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:watcherSubscription xmlns:pr="urn:oma:xml:rest:presence:1">
  <presentityUserId>tel:+1-555-100</presentityUserId>
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/watchersNotification</notifyURL>
    <callbackData>1234</callbackData>
  </callbackReference>
  <clientCorrelator>321</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <duration>7200</duration>
```

```
  <resourceStatusFilter>Pending</resourceStatusFilter>
  <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/subscriptions/watchersSubscriptions/sub001
  </resourceURL>
</pr:watcherSubscription>
```

C.4 Create Presence subscription

This operation is used by a Watcher to create a new Presence subscription towards the specified Presentity.

The request parameters are as follows:

Name	Type/value	Optional	Description
presentityUserId	xsd:anyURI	Yes	Identifies the Presentity for which the subscription is created towards. Mandatory in responses
notifyURL	xsd:anyURI	No	Notification endpoint definition
callbackData	xsd:string	Yes	Data the application can register with the server when subscribing to notifications, and that are passed back unchanged in each of the related notifications.
notificationFormat	common:NotificationFormat	Yes	Default: XML Application can specify format of the resource representation in notifications that are related to this subscription. The choice is between {XML, JSON}.
clientCorrelator	xsd:string	Yes	A correlator that the client MAY use to tag this particular resource representation during a request to create a resource on the server. In case the field is present, the server SHALL not alter its value, and SHALL provide it as part of the representation of this resource. In case the field is not present, the server SHALL NOT generate it.
applicationTag	xsd:string	Yes	A tag that the client MAY use to tag this particular resource on the server. In case the field is present, the server SHALL not alter its value, and SHALL provide it as part of the representation of this resource. In case the field is not present, the server SHALL NOT generate it.
anonymous	xsd:boolean	Yes	Allows the Watcher to request that its user identity is not revealed to the Presentity. XSD modeling sets this parameter value to “true”
duration	xsd:int	Yes	Specifies the duration of the subscription in seconds. When this time has elapsed the subscription will expire unless it has been refreshed (by using some other methods than form-urlencoded).
presenceFilter	xsd:anyURI [0..unbounded]	Yes	Allows the Watcher to indicate what presence data it is interested of. The desired attributes are

			indicated with relative paths according to the [ResourceRelPath] in sections 5.2.3, 5.2.4, 5.2.5 and 5.2.6. If the parameter is omitted or there is an empty filter it means monitoring of all attribute types.
frequency	xsd:int	Yes	Maximum frequency of notifications, expressed as minimum time between notifications in seconds.

If the operation was successful, it returns an HTTP Status of "201 Created".

C.4.1 Example: creating new Presence subscription for Presentity (Informative)

C.4.1.1 Request

```
POST /exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceSubscriptions/tel%3A%2B1-555-100 HTTP/1.1
Accept: application/xml
Host: example.com:80
Content-Type: application/x-www-form-urlencoded
Content-Length: nnnn

presentityUserId=tel%3A%2B1-555-100&
notifyURL= http://application.example.com/notifications/presenceNotification&
callbackData=1234&
clientCorrelator=321&
applicationTag=myApp&
duration=7200
```

C.4.1.2 Response

```
HTTP/1.1 201 Created
Location: http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceSubscriptions/tel%3A%2B1-555-100/sub001
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<pr:presenceSubscription xmlns:pr="urn:oma:xml:rest:presence:1">
  <presentityUserId>tel:+1-555-100</presentityUserId>
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/presenceNotification</notifyURL>
    <callbackData>1234</callbackData>
  </callbackReference>
  <clientCorrelator>321</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <duration>7200</duration>
  <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceSubscriptions/tel%3A%2B1-555-100/sub001
  </resourceURL>
</pr:presenceSubscription>
```

C.5 Create Presence list subscription

This operation is used by a Watcher to create a new Presence list subscription towards the specified Presence list.

The request parameters are as follows:

Name	Type/values	Optional	Description
presenceListId	xsd:string	Yes	Identifies the Presence list for which the subscription is created towards. Mandatory in responses.
notifyURL	xsd:anyURI	No	Notification endpoint definition
callbackData	xsd:string	Yes	Data the application can register with the server when subscribing to notifications, and that are passed back unchanged in each of the related notifications.
notificationFormat	common:NotificationFormat	Yes	Default: XML Application can specify format of the resource representation in notifications that are related to this subscription. The choice is between {XML, JSON}.
clientCorrelator	xsd:string	Yes	A correlator that the client MAY use to tag this particular resource representation during a request to create a resource on the server. In case the field is present, the server SHALL not alter its value, and SHALL provide it as part of the representation of this resource. In case the field is not present, the server SHALL NOT generate it.
applicationTag	xsd:string	Yes	A tag that the client MAY use to tag this particular resource on the server. In case the field is present, the server SHALL not alter its value, and SHALL provide it as part of the representation of this resource. In case the field is not present, the server SHALL NOT generate it.
anonymous	xsd:boolean	Yes	Allows the Watcher to request that its user identity is not revealed to the Presentity. XSD modeling sets this parameter value to “true”
duration	xsd:int	Yes	Specifies the duration of the subscription in seconds. When this time has elapsed the subscription will expire unless it has been refreshed (by using some other methods than form-urlencoded).
presenceFilter	xsd:anyURI [0..unbounded]	Yes	Allows the Watcher to indicate what presence data it is interested of. The desired attributes are indicated with relative paths according to the [ResourceRelPath] in sections 5.2.3, 5.2.4, 5.2.5 and 5.2.6. If the parameter is omitted or there is an empty filter it means monitoring of all attribute types.

frequency	xsd:int	Yes	Maximum frequency of notifications, expressed as minimum time in seconds between notifications in seconds.

If the operation was successful, it returns an HTTP Status of "201 Created".

C.5.1 Example: creating new Presence list subscription towards a single Presence list (Informative)

C.5.1.1 Request

```
POST /exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceListSubscriptions/myFriends HTTP/1.1
Accept: application/xml
Host: example.com:80
Content-Type: application/x-www-form-urlencoded
Content-Length: nnnn

presenceListId=myFriends&
notifyURL=http://application.example.com/notifications/presenceListNotification&
callbackData=1234&
clientCorrelator=321&
applicationTag=myApp&
duration=7200&
presenceFilter=person%2fmood&
frequency=600
```

C.5.1.2 Response

```
HTTP/1.1 201 Created
Location: http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceListSubscriptions/myFriends/sub001
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<pr:presenceListSubscription xmlns:pr="urn:oma:xml:rest:presence:1">
  <presenceListId>myFriends</presenceListId>
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/presenceListNotification</notifyURL>
    <callbackData>1234</callbackData>
  </callbackReference>
  <clientCorrelator>321</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <duration>5200</duration>
  <presenceFilter>person/mood</presenceFilter>
  <frequency>600</frequency>
  <resourceURL>http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceListSubscriptions/myFriends/
   sub001</resourceURL>
</pr:presenceListSubscription>
```

Appendix D. JSON examples (Informative)

JSON (JavaScript Object Notation) is a lightweight, 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 Parlay REST invocations from browsers or other processors with JavaScript engines. Further information on JSON can be found at [RFC 4627].

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

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

D.1 Retrieving all presence sources for user (section 5.4.3.1)

Request:

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources HTTP/1.1
Host: example.com:80
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"presenceSourceList": {"presenceSource": [
   {
      "applicationTag": "myApp",
      "clientCorrelator": "123",
      "duration": "3575",
      "presence": {
         "device": {
            "deviceId": "mac:321",
            "networkAvailability": {"network": {
               "connectionStatus": "Active",
               "id": "GPRS"
            }}
         },
         "person": {"mood": {"moodValue": "Happy"}},
         "service": {
            "devices": {"deviceId": "mac:321"},
            "serviceAvailability": "Open",
            "serviceId": "org.openmobilealliance:IM-Session",
            "version": "1.0"
         }
      },
      "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/prs123"
   },
   {
      "presence": {"person": {"noteList": {"note": {
         "$t": "I am on vacation!",
         "lang": "en"
```

```
        }}}},
        "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/persistent"
    }
]}}
```

D.2 Retrieving of all presence sources Meta data using a filter (section 5.4.3.2)

Request:

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources?presenceSourceFilter=presenceSourceMetaData HTTP/1.1
Host: example.com:80
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"presenceSourceList": {"presenceSource": [
    {
        "applicationTag": "myApp",
        "clientCorrelator": "123",
        "duration": "3575",
        "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/prs123"
    },
    {"resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/persistent"}
]}}
        "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/persistent"
    }
]}}
```

D.3 Creating presence source for user (section 5.4.5.1)

Request:

```
POST /exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources HTTP/1.1
Host: example.com:80
Content-Type: application/json
Accept: application/json
Content-Length: nnnn

{"presenceSource": {
    "applicationTag": "myApp",
    "clientCorrelator": "123",
    "duration": "7200",
    "presence": {
```

```
      "device": {
         "deviceId": "mac:321",
         "networkAvailability": {"network": {
            "connectionStatus": "Active",
            "id": "GPRS"
         }}
      },
      "person": {"mood": {"moodValue": "Happy"}},
      "service": {
         "devices": {"deviceId": "mac:321"},
         "serviceAvailability": "Open",
         "serviceId": "org.openmobilealliance:IM-Session",
         "version": "1.0"
      }
   }
}}
```

Response:

```
HTTP/1.1 201 Created
Location: http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/prs123
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"presenceSource": {
   "applicationTag": "myApp",
   "clientCorrelator": "123",
   "duration": "7200",
   "presence": {
      "device": {
         "deviceId": "mac:321",
         "networkAvailability": {"network": {
            "connectionStatus": "Active",
            "id": "GPRS"
         }}
      },
      "person": {"mood": {"moodValue": "Happy"}},
      "service": {
         "devices": {"deviceId": "mac:321"},
         "serviceAvailability": "Open",
         "serviceId": "org.openmobilealliance:IM-Session",
         "version": "1.0"
      }
   },
   "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/prs123"
}}
```

D.4 Creating presence source for user fails (section 5.4.5.2)

Request:

```
POST /exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources HTTP/1.1
```

```
Content-Type: application/json
Accept: application/json
Content-Length: nnnn
Host: example.com:80

{"presenceSource": {
   "applicationTag": "myApp",
   "clientCorrelator": "123",
   "duration": "7200",
   "presence": {"person": {"mood": {"moodValue": "Happy"}}}
}}
```

Response:

```
HTTP/1.1 409 Conflict
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"requestError": {
   "link": {
      "href": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources",
      "rel": "PresenceSourceList"
   },
   "policyException": {
      "messageId": "POL0001",
      "text": "A policy error occurred. Error code is %1",
      "variables": "Max number of presence sources reached"
   }
}}
```

D.5 Retrieving presence source (section 5.5.3.1)

Request:

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/prs123 HTTP/1.1
Host: example.com:80
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"presenceSource": {
   "applicationTag": "myApp",
   "clientCorrelator": "123",
   "duration": "5237",
   "presence": {
      "device": {
         "deviceId": "mac:321",
```

```
            "networkAvailability": {"network": {
               "connectionStatus": "Active",
               "id": "GPRS"
            }}
         },
         "person": {"mood": {"moodValue": "Happy"}},
         "service": {
            "devices": {"deviceId": "mac:321"},
            "serviceAvailability": "Open",
            "serviceId": "org.openmobilealliance:IM-Session",
            "version": "1.0"
         }
      },
      "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/prs123"
}}
```

D.6 Retrieving presence source which does not exist (section 5.5.3.2)

Request:

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/prs123 HTTP/1.1
Host: example.com:80
Accept: application/json
```

Response:

```
HTTP/1.1 404 Not Found
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"requestError": {
   "link": {
      "href": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/prs123",
      "rel": "PresenceSourceList"
   },
   "serviceException": {
      "messageId": "SVC0001",
      "text": "A service error occurred. Error code is %1",
      "variables": "Presence source does not exist"
   }
}}
```

D.7 Updating presence source (section 5.5.4.1)

Request:

```
PUT /exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/prs123 HTTP/1.1
Content-Type: application/json
```

```
Accept: application/json
Content-Length: nnnn
Host: example.com:80

{"presenceSource": {
   "duration": "7200",
   "presence": {
      "device": {
         "deviceId": "mac:321",
         "networkAvailability": {"network": {
            "connectionStatus": "Active",
            "id": "GPRS"
         }}
      },
      "person": {"mood": {"moodValue": "Invincible"}},
      "service": {
         "devices": {"deviceId": "mac:321"},
         "serviceAvailability": "Closed",
         "serviceId": "org.openmobilealliance:IM-Session",
         "version": "1.0"
      }
   }
}}
```

Response:

```
HTTP/1.1 201 Created
Location: http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/{presenceSourceId}
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"presenceSource": {
   "duration": "7200",
   "presence": {
      "device": {
         "deviceId": "mac:321",
         "networkAvailability": {"network": {
            "connectionStatus": "Active",
            "id": "GPRS"
         }}
      },
      "person": {"mood": {"moodValue": "Invincible"}},
      "service": {
         "devices": {"deviceId": "mac:321"},
         "serviceAvailability": "Closed",
         "serviceId": "org.openmobilealliance:IM-Session",
         "version": "1.0"
      }
   },
   "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/prs123"
}}
```

D.8 Removing presence source (section 5.5.6.1)

Request:

```
DELETE /exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/prs123 HTTP/1.1
Host: example.com:80
```

Response:

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

D.9 Retrieving individual presence attribute (section 5.6.3.1)

Request:

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/prs123/person/mood HTTP/1.1
Host: example.com:80
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"mood": {"moodValue": "Happy"}}
```

D.10 Updating individual presence attribute (section 5.6.4.1)

Request:

```
PUT /exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/prs123/person/mood HTTP/1.1
Content-Type: application/json
Accept: application/json
Content-Length: nnnn
Host: example.com:80

{"mood": {"moodValue": "Excited"}}
```

Response:

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"mood": {"moodValue": "Excited"}}
```

D.11 Removing individual presence attribute (section 5.6.6.1)

Request:

```
DELETE /exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/prs123/person/mood HTTP/1.1
Host: example.com:80
```

Response:

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

D.12 Retrieving persistent presence data (section 5.7.3.1)

Request:

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/persistent?resFormat=JSON HTTP/1.1
Host: example.com:80
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"presenceSource": {
   "presence": {"person": {"noteList": {"note": {
      "$t": "Im on vacation!",
      "lang": "en"
   }}}},
   "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/persistent"
}}
```

D.13 Updating persistent presence data (section 5.7.4.1)

Request:

```
PUT /exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/persistent HTTP/1.1
Host: example.com:80
If-Match: "10"
Content-Type: application/json
Accept: application/json
Content-Length: nnnn

{"presenceSource": {"presence": {"person": {
   "noteList": {"note": {
      "$t": "My picture is updated!",
      "lang": "en"
   }},
```

```
   "statusIcon": {
      "contentType": "image/jpg",
      "eTag": "123",
      "statusIconAddress": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/content/pic001.jpg"
   }
}}}}
```

Response:

```
HTTP/1.1 412 Precondition Failed
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

Request:

```
PUT /exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/persistent HTTP/1.1
Host: example.com:80
If-Match: "11"
Content-Type: application/json
Accept: application/json
Content-Length: nnnn

{"presenceSource": {"presence": {"person": {
   "noteList": {"note": {
      "$t": "My picture is updated!",
      "lang": "en"
   }},
   "statusIcon": {
      "contentType": "image/jpg",
      "eTag": "123",
      "statusIconAddress": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/content/pic001.jpg"
   }
}}}}
```

Response:

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
ETag: "12"
Content-Type: application/json
Content-Length: nnnn

{"presenceSource": {
   "presence": {"person": {
      "noteList": {"note": {
         "$t": "My picture is updated!",
         "lang": "en"
      }},
      "statusIcon": {
         "contentType": "image/jpg",
         "eTag": "123",
```

```
            "statusIconAddress": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/content/pic001.jpg"
        }
    }},
    "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/persistent"
}}
```


D.14 Removing persistent presence data (section 5.7.6.1)

Request:

```
DELETE /exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/persistent HTTP/1.1
Host: example.com:80
```

Response:

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

D.15 Retrieving individual persistent presence attribute (section 5.8.3.1)

Request:

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/persistent/person/statusIcon HTTP/1.1
Host: example.com:80
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"statusIcon": {
    "contentType": "image/jpg",
    "eTag": "123",
    "statusIconAddress": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/content/pic001.jpg"
}}
```

D.16 Updating individual persistent presence attribute (section 5.8.4.1)

Request:

```
PUT /exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/persistent/person/statusIcon HTTP/1.1
```

```
Host: example.com:80
Content-Type: application/json
Accept: application/json
Content-Length: nnnn

{"statusIcon": {
   "contentType": "image/jpg",
   "eTag": "456",
   "statusIconAddress": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/content/pic001.jpg"
}}
```

Response:

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"statusIcon": {
   "contentType": "image/jpg",
   "eTag": "456",
   "statusIconAddress": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/content/pic001.jpg"
}}
```

D.17 Removing individual persistent presence attribute (section 5.8.6.1)

Request:

```
DELETE /exampleAPI/1/presence/tel%3A%2B1-555-100/presenceSources/persistent/person/mood HTTP/1.1
Host: example.com:80
```

Response:

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

D.18 Retrieving list of available contents (section 5.9.3.1)

Request:

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-100/content HTTP/1.1
Host: example.com:80
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"contentList": {
   "content": [
      {
         "contentType": "image/jpg",
         "link": {
            "href": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/content/pic003.jpg",
            "rel": "content"
         }
      },
      {
         "contentType": "image/jpg",
         "link": {
            "href": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/content/pic004.jpg",
            "rel": "content"
         }
      }
   ],
   "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/content"
}}
```

D.19 Retrieving individual content by Presentity (section 5.10.3.1)

Request:

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-100/content/pic001.jpg HTTP/1.1
Host: example.com:80
Accept: image/jpg
```

Response:

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: image/jpg
Content-Length: nnnn

data
```

D.20 Uploading/updating individual content by Presentity (section 5.10.4.1)

Request:

```
PUT /exampleAPI/1/presence/tel%3A%2B1-555-100/content/pic001.jpg HTTP/1.1
Host: example.com:80
Content-Type: image/jpg
Content-Length: nnnn

data
```

Response:

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

D.21 Removing individual content by Presentity (section 5.10.6.1)

Request:

```
DELETE /exampleAPI/1/presence/tel%3A%2B1-555-100/content/pic001.jpg HTTP/1.1
Host: example.com:80
```

Response:

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

D.22 Retrieving list of Watchers (section 5.11.3.1)

Request:

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-100/watchers?resourceStatusFilter=Pending HTTP/1.1
Host: example.com:80
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"watcherList": {
   "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/watchers",
   "watcher": {
      "displayName": "Bob",
      "resourceStatus": "Pending",
      "watcherUserId": "tel:+1-555-101"
   }
}}
```

D.23 Retrieving individual Watcher (section 5.12.3.1)

Request:

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-100/watchers/{tel%3A%2B1-555-101 HTTP/1.1
Host: example.com:80
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"watcher": {
   "displayName": "Bob",
   "resourceStatus": "Pending",
   "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/watchers/tel%3A%2B1-555-101",
   "watcherUserId": "tel:+1-555-101"
}}
```

D.24 Retrieving all authorization rules (section 5.13.3.1)

Request:

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-100/authorization/rules HTTP/1.1
Host: example.com:80
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"ruleList": {
   "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/authorization/rules",
   "rule": [
      {
         "decision": "Allow",
         "ruleName": "allowList",
         "watcherUserId": [
            "tel:+1-555-102",
            "tel:+1-555-104"
         ]
      },
      {
         "decision": "Block",
         "memberListId": "myBlockList",
         "ruleName": "blockList"
      }
   ]
}}
```

D.25 Creating an authorization rule (section 5.13.5.1)

Request:

```
POST /exampleAPI/1/presence/tel%3A%2B1-555-100/authorization/rules HTTP/1.1
Host: example.com:80
Content-Type: application/json
Accept: application/json
Content-Length: nnnn

{"rule": {
   "decision": "Confirm",
   "otherUser": null,
   "ruleName": "otherUsers"
}}
```

Response:

```
HTTP/1.1 201 Created
Location: http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/authorization/rules/rule003
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"rule": {
   "decision": "Confirm",
   "otherUser": null,
   "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/authorization/rules/rule003",
   "ruleName": "otherUsers"
}}
```

D.26 Creating an authorization rule, response with resourceReference (section 5.13.5.2)

Request:

```
POST /exampleAPI/1/presence/tel%3A%2B1-555-100/authorization/rules HTTP/1.1
Host: example.com:80
Content-Type: application/json
Accept: application/json
Content-Length: nnnn

{"rule": {
   "decision": "Confirm",
   "otherUser": null,
   "ruleName": "otherUsers"
}}
```

Response:

```
HTTP/1.1 201 Created
Location: http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/authorization/rules/rule003
```

```
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"resourceReference": {"resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/authorization/rules/rule003"}}
```

D.27 Retrieving an authorization rule (section 5.14.3.1)

Request:

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-100/authorization/rules/rule001 HTTP/1.1
Host: example.com:80
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"rule": {
   "decision": "Allow",
   "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/authorization/rules/rule001",
   "ruleName": "allowList",
   "watcherUserId": [
      "tel:+1-555-102",
      "tel:+1-555-104"
   ]
}}
```

D.28 Updating an authorisation rule (section 5.14.4.1)

Request:

```
PUT /exampleAPI/1/presence/tel%3A%2B1-555-100/authorization/rules/rule001 HTTP/1.1
Host: example.com:80
Content-Type: application/json
Accept: application/json
Content-Length: nnnn

{"rule": {
   "decision": "Allow",
   "ruleName": "allowList",
   "watcherUserId": [
      "tel:+1-555-102",
      "tel:+1-555-104",
      "tel:+1-555-105"
   ]
}}
```

Response:

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"rule": {
   "decision": "Allow",
   "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/authorization/rules/rule001",
   "ruleName": "allowList",
   "watcherUserId": [
      "tel:+1-555-102",
      "tel:+1-555-104",
      "tel:+1-555-105"
   ]
}}
```

D.29 Removing an authorisation rule (section 5.14.6.1)

Request:

```
DELETE /exampleAPI/1/presence/tel%3A%2B1-555-100/authorization/rules/rule002 HTTP/1.1
Host: example.com:80
```

Response:

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

D.30 Retrieving individual authorization rule data (section 5.15.3.1)

Request:

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-100/authorization/rules/rule002/watchers/tel%3A%2B1-555-102 HTTP/1.1
Host: example.com:80
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"watcherUserId": "\n    tel:+1-555-102\n"}
```

D.31 Updating individual authorization rule data (section 5.15.4.1)

Request:

```
PUT /exampleAPI/1/presence/tel%3A%2B1-555-100/authorization/rules/rule002/watchers/tel%3A%2B1-555-103 HTTP/1.1
Host: example.com:80
Content-Type: application/json
Accept: application/json
Content-Length: nnnn

{"watcherUserId": "\n       tel:+1-555-103\n"}
```

Response:

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"watcherUserId": "\n      tel:+1-555-103\n"}
```

D.32 Removing individual authorization rule data (section 5.15.6.1)

Request:

```
DELETE /exampleAPI/1/presence/tel%3A%2B1-555-100/authorization/rules/rule002/watchers/tel%3A%2B1-555-103
Host: example.com:80
```

Response:

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

D.33 Retrieving all presence data for Presentity (section 5.16.3.1)

Request:

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-101/presenceContacts/tel%3A%2B1-555-100 HTTP/1.1
Host: example.com:80
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"presenceContact": {
```

```
   "presence": {
      "device": {
         "deviceId": "mac:321",
         "networkAvailability": {"network": {
            "connectionStatus": "Active",
            "id": "GPRS"
         }}
      },
      "person": {
         "mood": {"moodValue": "Happy"},
         "noteList": {"note": {
            "$t": "Im on vacation!",
            "lang": "en"
         }}
      },
      "service": {
         "devices": {"deviceId": "mac:321"},
         "serviceAvailability": "Open",
         "serviceId": "org.openmobilealliance:IM-Session",
         "version": "1.0"
      }
   },
   "presentityUserId": "tel:+1-555-100",
   "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/presenceContacts/tel%3A%2B1-555-100"
}}
```

D.34 Retrieving presence data for Presentity by using filter (section 5.16.3.2)

Request:

```
GET  /exampleAPI/1/presence/tel%3A%2B1-555-101/presenceContacts/tel%3A%2B1-555-
100?presenceFilter=person/mood&presenceFilter=service/org.openmobilealliance:IM-Session/1.0/serviceAvailability HTTP/1.1
Host: example.com:80
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"presenceContact": {
   "presence": {
      "person": {"mood": {"moodValue": "Happy"}},
      "service": {
         "devices": {"deviceId": "mac:321"},
         "serviceAvailability": "Open",
         "serviceId": "org.openmobilealliance:IM-Session",
         "version": "1.0"
```

```
      }
   },
   "presentityUserId": "tel:+1-555-100",
   "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/presenceContacts/tel%3A%2B1-555-100"
}}
```

D.35 Retrieving individual presence attribute for Presentity (section 5.17.3.1)

Request:

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-101/presenceContacts/tel%3A%2B1-555-100/person/noteList HTTP/1.1
Host: example.com:80
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"presenceContact": {
   "presence": {"person": {"noteList": {"note": {
      "$t": "Im on vacation!",
      "lang": "en"
   }}}},
   "presentityUserId": "tel:+1-555-100"
}}
```

D.36 Retrieving presence data for all Presentities in a Presence list (section 5.18.3.1)

Request:

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-101/presenceLists/myFriends HTTP/1.1
Host: example.com:80
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"presenceList": {
   "presenceContact": [
      {
```

```
        "presence": {
           "device": {
              "deviceId": "mac:321",
              "networkAvailability": {"network": {
                 "connectionStatus": "Active",
                 "id": "GPRS"
              }}
           },
           "person": {
              "mood": {"moodValue": "Happy"},
              "noteList": {"note": {
                 "$t": "Im on vacation!",
                 "lang": "en"
              }}
           },
           "service": {
              "devices": {"deviceId": "mac:321"},
              "serviceAvailability": "Open",
              "serviceId": "org.openmobilealliance:IM-Session",
              "version": "1.0"
           }
        },
        "presentityUserId": "tel:+1-555-100",
        "resourceStatus": "Active"
     },
     {
        "presentityUserId": "tel:+1-555-102",
        "resourceStatus": "Pending"
     },
     {
        "presentityUserId": "tel:+1-555-104",
        "resourceStatus": "TerminatedNoResource"
     }
  ],
  "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/presenceLists/myFriends"
}}
```


D.37 Retrieving content by Watcher (section 5.19.3.1)

Request:

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-101/presenceContactsContent/{tel%3A%2B1-555-100}/pic001.jpg HTTP/1.1
Host: example.com:80
Accept: image/jpg
```

Response:

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: image/jpg
Content-Length: nnnn

data
```

D.38 Retrieving all active subscriptions for user (section 5.20.3.1)

Request:

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions HTTP/1.1
Host: example.com:80
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"subscriptionList": {
   "presenceListSubscriptionCollection": {
      "presenceListSubscription": {
         "applicationTag": "myApp",
         "callbackReference": {
            "callbackData": "2345",
            "notifyURL": "http://application.example.com/notifications/presenceListNotification"
         },
         "clientCorrelator": "432",
         "duration": "4629",
         "presenceListId": "myFriends",
         "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-
101/subscriptions/presenceListSubscriptions/myFriends/\n      sub002"
      },
      "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceListSubscriptions"
   },
   "presenceSubscriptionList": {
      "presenceSubscription": {
         "applicationTag": "myApp",
         "callbackReference": {
            "callbackData": "1234",
            "notifyURL": "http://application.example.com/notifications/presenceNotification"
         },
         "clientCorrelator": "321",
         "duration": "5246",
         "presentityUserId": "tel:+1-555-100",
         "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-
101/subscriptions/presenceSubscriptions/tel%3A%2B1-555-100/sub001\n      "
      },
      "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceSubscriptions"
   },
   "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions",
   "watchersSubscriptionList": {
      "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/watchersSubscriptions",
      "watcherSubscription": {
         "applicationTag": "myApp",
         "callbackReference": {
            "callbackData": "3456",
            "notifyURL": "http://application.example.com/notifications/watchersNotification"
         },
         "clientCorrelator": "543",
```

```
        "duration": "2413",
        "presentityUserId": "tel:+1-555-100",
        "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/watchersSubscriptions/sub003\n
"
      }
   }
}}
```

D.39 Retrieving all Watchers subscriptions (section 5.21.3.1)

Request:

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-100/subscriptions/watchersSubscriptions HTTP/1.1
Host: example.com:80
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"watchersSubscriptionList": {
   "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/subscriptions/watchersSubscriptions",
   "watcherSubscription": [
      {
         "applicationTag": "myApp",
         "callbackReference": {
            "callbackData": "1234",
            "notifyURL": "http://application.example.com/notifications/watchersNotification"
         },
         "clientCorrelator": "321",
         "duration": "5246",
         "presentityUserId": "tel:+1-555-100",
         "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-
100/subscriptions/watchersSubscriptions/tel%3A%2B1-555-100/sub001\n    "
      },
      {
         "applicationTag": "myApp",
         "callbackReference": {
            "callbackData": "4321",
            "notifyURL": "http://application.example.com/notifications/watchersNotification"
         },
         "clientCorrelator": "123",
         "duration": "5237",
         "presentityUserId": "tel:+1-555-100",
         "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-
100/subscriptions/watchersSubscriptions/tel%3A%2B1-555-100/sub002\n    "
      }
   ]
}}
```

D.40 Creating new Watchers subscription (section 5.21.5.1)

Request:

```
POST /exampleAPI/1/presence/tel%3A%2B1-555-100/subscriptions/watchersSubscriptions HTTP/1.1
Host: example.com:80
Content-Type: application/json
Accept: application/json
Content-Length: nnnn

{"watcherSubscription": {
   "applicationTag": "myApp",
   "callbackReference": {
      "callbackData": "1234",
      "notifyURL": "http://application.example.com/notifications/watchersNotification"
   },
   "clientCorrelator": "321",
   "duration": "7200",
   "presentityUserId": "tel:+1-555-100"
}}
```

Response:

```
HTTP/1.1 201 Created
Location: http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/subscriptions/watchersSubscriptions/sub001
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"watcherSubscription": {
   "applicationTag": "myApp",
   "callbackReference": {
      "callbackData": "1234",
      "notifyURL": "http://application.example.com/notifications/watchersNotification"
   },
   "clientCorrelator": "321",
   "duration": "3600",
   "presentityUserId": "tel:+1-555-100",
   "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/subscriptions/watchersSubscriptions/sub001\n  "
}}
```

D.41 Retrieving individual Watchers subscription (section 5.22.3.1)

Request:

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-100/subscriptions/watchersSubscriptions/sub001 HTTP/1.1
Host: example.com:80
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"watcherSubscription": {
   "applicationTag": "myApp",
   "callbackReference": {
      "callbackData": "1234",
      "notifyURL": "http://application.example.com/notifications/watchersNotification"
   },
   "clientCorrelator": "321",
   "duration": "5246",
   "presentityUserId": "tel:+1-555-100",
   "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/subscriptions/watchersSubscriptions/sub001\n  "
}}
```

D.42 Updating individual Watchers subscription (section 5.22.4.1)

Request:

```
PUT /exampleAPI/1/presence/tel%3A%2B1-555-100/subscriptions/watchersSubscriptions/sub001 HTTP/1.1
Host: example.com:80
Content-Type: application/json
Accept: application/json
Content-Length: nnnn

{"watcherSubscription": {
   "applicationTag": "myApp",
   "callbackReference": {
      "callbackData": "1234",
      "notifyURL": "http://application.example.com/notifications/watchersNotification"
   },
   "clientCorrelator": "321",
   "duration": "7200",
   "presentityUserId": "tel:+1-555-100"
}}
```

Response:

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"watcherSubscription": {
   "applicationTag": "myApp",
   "callbackReference": {
      "callbackData": "1234",
      "notifyURL": "http://application.example.com/notifications/watchersNotification"
   },
   "clientCorrelator": "321",
   "duration": "7200",
```

```
   "presentityUserId": "tel:+1-555-100",
   "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/subscriptions/watchersSubscriptions/sub001\n  "
}}
```

D.43 Terminating individual Watchers subscription (section 5.22.6.1)

Request:

```
DELETE /exampleAPI/1/presence/tel%3A%2B1-555-100/subscriptions/watchersSubscriptions/sub001 HTTP/1.1
Host: example.com:80
```

Response:

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

D.44 Notifying Presentity about change in Watchers status (section 5.23.5.1)

Request:

```
POST /notifications/watchersNotification HTTP/1.1
Host: example.com:80
Content-Type: application/json
Accept: application/json
Content-Length: nnnn

{"watcherNotification": {
   "callbackData": "1234",
   "link": {
      "href": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/subscriptions/watchersSubscriptions/sub001",
      "rel": "WatcherSubscription"
   },
   "presentityUserId": "tel:+1-555-100",
   "resourceStatus": "Active",
   "watcherList": {"watcher": {
      "displayName": "Bob",
      "resourceStatus": "Pending",
      "watcherUserId": "tel:+1-555-101"
   }}
}}
```

Response:

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

D.45 Notifying Watcher about subscription time out (section 5.23.5.2)

Request:

```
POST /notifications/watchersNotification HTTP/1.1
Host: example.com:80
Content-Type: application/json
Accept: application/json
Content-Length: nnnn

{"watcherNotification": {
   "callbackData": "1234",
   "link": {
      "href": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/subscriptions/watchersSubscriptions/sub001",
      "rel": "WatcherSubscription"
   },
   "presentityUserId": "tel:+1-555-100",
   "resourceStatus": "TerminatedTimeout"
}}
```

Response:

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

D.46 Notifying Watcher about termination of Watchers subscription (reason unknown) (section 5.23.5.3)

Request:

```
POST /notifications/watchersNotification HTTP/1.1
Host: example.com:80
Content-Type: application/json
Accept: application/json
Content-Length: nnnn

{"watcherNotification": {
   "callbackData": "1234",
   "link": {
      "href": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-100/subscriptions/watchersSubscriptions/sub001",
      "rel": "WatcherSubscription"
   },
   "presentityUserId": "tel:+1-555-100",
   "resourceStatus": "TerminatedOther"
}}
```

Response:

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

D.47 Retrieving all Presence subscriptions for all Presentities (section 5.24.3.1)

Request:

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceSubscriptions HTTP/1.1
Host: example.com:80
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"presenceSubscriptionList": {
   "presenceSubscription": [
      {
         "applicationTag": "myApp",
         "callbackReference": {
            "callbackData": "1234",
            "notifyURL": "http://application.example.com/notifications/presenceNotification"
         },
         "clientCorrelator": "321",
         "duration": "5246",
         "presentityUserId": "tel:+1-555-100",
         "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-
101/subscriptions/presenceSubscriptions/tel%3A%2B1-555-100/sub001\n    "
      },
      {
         "applicationTag": "myApp",
         "callbackReference": {
            "callbackData": "4321",
            "notifyURL": "http://application.example.com/notifications/presenceNotification"
         },
         "clientCorrelator": "123",
         "duration": "5237",
         "presentityUserId": "tel:+1-555-100",
         "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-
101/subscriptions/presenceSubscriptions/tel%3A%2B1-555-101/sub004\n    "
      }
   ],
   "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceSubscriptions"
}}
```

D.48 Retrieving all Presence subscriptions for Presentity (section 5.25.3.1)

Request:

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceSubscriptions/tel%3A%2B1-555-100 HTTP/1.1
```

```
Host: example.com:80
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"presenceSubscriptionList": {
   "presenceSubscription": [
      {
         "applicationTag": "myApp",
         "callbackReference": {
            "callbackData": "1234",
            "notifyURL": "http://application.example.com/notifications/presenceNotification"
         },
         "clientCorrelator": "321",
         "duration": "5246",
         "presentityUserId": "tel:+1-555-100",
         "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-
101/subscriptions/presenceSubscriptions/tel%3A%2B1-555-100/sub001\n    "
      },
      {
         "applicationTag": "myApp",
         "callbackReference": {
            "callbackData": "6789",
            "notifyURL": "http://application2.example.com/notifications/presenceNotification"
         },
         "clientCorrelator": "987",
         "duration": "4132",
         "presentityUserId": "tel:+1-555-100",
         "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-
101/subscriptions/presenceSubscriptions/tel%3A%2B1-555-100/sub005\n    "
      }
   ],
   "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceSubscriptions/tel%3A%2B1-
555-100"
}}
```

D.49 Creating new Presence subscription for Presentity (section 5.25.5.1)

Request:

```
POST /exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceSubscriptions/tel%3A%2B1-555-100 HTTP/1.1
Host: example.com:80
Content-Type: application/json
Accept: application/json
Content-Length: nnnn
```

```
{"presenceSubscription": {
   "applicationTag": "myApp",
   "callbackReference": {
      "callbackData": "1234",
      "notifyURL": "http://application.example.com/notifications/presenceNotification"
   },
   "clientCorrelator": "321",
   "duration": "7200",
   "frequency": "600"
}}
```

Response:

```
HTTP/1.1 201 Created
Location: http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/presenceSubscriptions/tel%3A%2B1-555-100/sub001
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"presenceSubscription": {
   "applicationTag": "myApp",
   "callbackReference": {
      "callbackData": "1234",
      "notifyURL": "http://application.example.com/notifications/presenceNotification"
   },
   "clientCorrelator": "321",
   "duration": "7200",
   "frequency": "600",
   "presentityUserId": "tel:+1-555-100",
   "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceSubscriptions/tel%3A%2B1-
555-100/sub001\n  "
}}
```

D.50 Creating new Presence subscription for unknown Presentity (section 5.25.5.2)

Request:

```
POST /exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceSubscriptions/tel%3A%2B1-555-100 HTTP/1.1
Host: example.com:80
Content-Type: application/json
Accept: application/json
Content-Length: nnnn

{"presenceSubscription": {
   "applicationTag": "myApp",
   "callbackReference": {
      "callbackData": "1234",
      "notifyURL": "http://application.example.com/notifications/presenceNotification"
   },
   "clientCorrelator": "321",
   "duration": "7200",
```

```
   "frequency": "600"
}}
```

Response:

```
HTTP/1.1 201 Created
Location: http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/presenceSubscriptions/tel%3A%2B1-555-100/sub001
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"requestError": {
   "link": {
      "href": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/presenceSources",
      "rel": "PresenceSourceList"
   },
   "serviceException": {
      "messageId": "SVC0004",
      "text": "No valid addresses provided in message part %1",
      "variables": "The specified identity does not exists."
   }
}}
```

D.51 Retrieving individual Presence subscription (section 5.26.3.1)

Request:

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceSubscriptions/tel%3A%2B1-555-100/sub001 HTTP/1.1
Host: example.com:80
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"presenceSubscription": {
   "applicationTag": "myApp",
   "callbackReference": {
      "callbackData": "1234",
      "notifyURL": "http://application.example.com/notifications/presenceNotification"
   },
   "clientCorrelator": "321",
   "duration": "5246",
   "frequency": "600",
   "presentityUserId": "tel:+1-555-100",
   "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceSubscriptions/tel%3A%2B1-
555-100/sub001\n  "
}}
```

D.52 Updating individual Presence subscription (section 5.26.4.1)

Request:

```
PUT /exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceSubscriptions/tel%3A%2B1-555-100/sub001 HTTP/1.1
Host: example.com:80
Content-Type: application/json
Accept: application/json
Content-Length: nnnn

{"presenceSubscription": {
   "applicationTag": "myApp",
   "callbackReference": {
      "callbackData": "1234",
      "notifyURL": "http://application.example.com/notifications/presenceNotification"
   },
   "clientCorrelator": "321",
   "duration": "7200",
   "frequency": "600",
   "presentityUserId": "tel:+1-555-100"
}}
```

Response:

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"presenceSubscription": {
   "applicationTag": "myApp",
   "callbackReference": {
      "callbackData": "1234",
      "notifyURL": "http://application.example.com/notifications/presenceNotification"
   },
   "clientCorrelator": "321",
   "duration": "7200",
   "frequency": "600",
   "presentityUserId": "tel:+1-555-100",
   "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceSubscriptions/tel%3A%2B1-
555-100/sub001\n  "
}}
```

D.53 Terminating individual Presence subscription (section 5.26.6.1)

Request:

```
DELETE /exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceSubscriptions/tel%3A%2B1-555-100/sub001HTTP/1.1
Host: example.com:80
```

Response:

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

D.54 Notifying Watcher about Presence data updates from an active subscription (section 5.27.5.1)

Request:

```
POST /notifications/presenceNotification HTTP/1.1
Host: example.com:80
Content-Type: application/json
Accept: application/json
Content-Length: nnnn

{"presenceNotification": {
   "callbackData": "1234",
   "link": {
      "href": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceSubscriptions/tel%3A%2B1-555-
100/sub001",
      "rel": "PresenceSubscription"
   },
   "presence": {"person": {"mood": {"moodValue": "Happy"}}},
   "presentityUserId": "tel:+1-555-100",
   "resourceStatus": "Active"
}}
```

Response:

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

D.55 Notifying Watcher about Presence data updates from pending subscription (section 5.27.5.2)

Request:

```
POST /notifications/presenceNotification HTTP/1.1
Host: example.com:80
Content-Type: application/json
Accept: application/json
Content-Length: nnnn

{"presenceNotification": {
   "callbackData": "2345",
   "link": {
      "href": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceSubscriptions/tel%3A%2B1-555-
101/sub002",
      "rel": "PresenceSubscription"
   },
```

```
   "presentityUserId": "tel:+1-555-100",
   "resourceStatus": "Pending"
}}
```

Response:

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

D.56 Notifying Watcher about termination of Presence subscription (reason unknown) (section 5.27.5.3)

Request:

```
POST /notifications/presenceNotification HTTP/1.1
Host: example.com:80
Content-Type: application/json
Accept: application/json
Content-Length: nnnn

{"presenceNotification": {
   "callbackData": "1234",
   "link": {
      "href": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceSubscriptions/tel%3A%2B1-555-
100/sub001",
      "rel": "PresenceSubscription"
   },
   "presentityUserId": "tel:+1-555-100",
   "resourceStatus": "TerminatedOther"
}}
```

Response:

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

D.57 Notifying Watcher about termination of Presence subscription (Watcher blocked) (section 5.27.5.4)

Request:

```
POST /notifications/presenceNotification HTTP/1.1
Host: example.com:80
Content-Type: application/json
Accept: application/json
Content-Length: nnnn

{"presenceNotification": {
   "callbackData": "1234",
```

```
   "link": {
      "href": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceSubscriptions/tel%3A%2B1-555-
100/sub001",
      "rel": "PresenceSubscription"
   },
   "presentityUserId": "tel:+1-555-100",
   "resourceStatus": "TerminatedBlocked"
}}
```

Response:

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

D.58 Retrieving all Presence list subscriptions towards all Presence lists (section 5.28.3.1)

Request:

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceListSubscriptions HTTP/1.1
Host: example.com:80
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"presenceListSubscriptionCollection": {
   "presenceListSubscription": [
      {
         "applicationTag": "myApp",
         "callbackReference": {
            "callbackData": "1234",
            "notifyURL": "http://application.example.com/notifications/presenceListNotification"
         },
         "clientCorrelator": "321",
         "duration": "5246",
         "presenceListId": "myFriends",
         "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-
101/subscriptions/presenceListSubscriptions/myFriends/\n     sub001"
      },
      {
         "applicationTag": "myApp",
         "callbackReference": {
            "callbackData": "4321",
            "notifyURL": "http://application.example.com/notifications/presenceListNotification"
         },
         "clientCorrelator": "123",
         "duration": "5237",
```

```
          "presenceListId": "myColleagues",
          "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-
101/subscriptions/presenceListSubscriptions/myColleagues/\n      sub002"
       }
   ],
   "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceListSubscriptions"
}}
```

D.59 Retrieving all Presence list subscriptions towards a single Presence list (section 5.29.3.1)

Request:

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceListSubscriptions/myFriends HTTP/1.1
Host: example.com:80
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"presenceListSubscriptionCollection": {
   "presenceListSubscription": {
      "applicationTag": "myApp",
      "callbackReference": {
         "callbackData": "1234",
         "notifyURL": "http://application.example.com/notifications/presenceListNotification"
      },
      "clientCorrelator": "321",
      "duration": "5246",
      "presenceListId": "myFriends",
      "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-
101/subscriptions/presenceListSubscriptions/myFriends/\n     sub001"
   },
   "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceListSubscriptions"
}}
```

D.60 Creating new Presence list subscription towards a single Presence list (section 5.29.5.1)

Request:

```
POST /exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceListSubscriptions/myFriends HTTP/1.1
Host: example.com:80
Content-Type: application/json
Accept: application/json
```

```
Content-Length: nnnn

{"presenceListSubscription": {
   "applicationTag": "myApp",
   "callbackReference": {
      "callbackData": "1234",
      "notifyURL": "http://application.example.com/notifications/presenceListNotification"
   },
   "clientCorrelator": "321",
   "duration": "7200",
   "frequency": "600"
   "presenceFilter": [
      "person/mood",
      "service/org.openmobilealliance:IM-Session/1.0"
   ],
}}
```

Response:

```
HTTP/1.1 201 Created
Location: http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceListSubscriptions/myFriends/sub001
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"presenceListSubscription": {
   "applicationTag": "myApp",
   "callbackReference": {
      "callbackData": "1234",
      "notifyURL": "http://application.example.com/notifications/presenceListNotification"
   },
   "clientCorrelator": "321",
   "duration": "7200",
   "frequency": "600",
   "presenceFilter": [
      "person/mood",
      "service/org.openmobilealliance:IM-Session/1.0"
   ],
   "presenceListId": "myFriends",
   "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-
101/subscriptions/presenceListSubscriptions/myFriends/\n   sub001"
}}
```

D.61 Retrieving individual Presence list subscription (section 5.30.3.1)

Request:

```
GET /exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceListSubscriptions/myFriends/sub001 HTTP/1.1
Host: example.com:80
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"presenceListSubscription": {
   "applicationTag": "myApp",
   "callbackReference": {
      "callbackData": "1234",
      "notifyURL": "http://application.example.com/notifications/presenceListNotification"
   },
   "clientCorrelator": "321",
   "duration": "5274",
   "frequency": "600",
   "presenceFilter": [
      "person/mood",
      "service/org.openmobilealliance:IM-Session/1.0"
   ],
   "presenceListId": "myFriends",
   "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-
101/subscriptions/presenceListSubscriptions/myFriends/\n   sub001"
}}
```

D.62 Updating individual Presence list subscription (section 5.30.4.1)

Request:

```
PUT /exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceListSubscriptions/myFriends/sub001 HTTP/1.1
Host: example.com:80
Content-Type: application/json
Accept: application/json
Content-Length: nnnn

{"presenceListSubscription": {
   "applicationTag": "myApp",
   "callbackReference": {
      "callbackData": "1234",
      "notifyURL": "http://application.example.com/notifications/presenceListNotification"
   },
   "clientCorrelator": "321",
   "duration": "7200",
   "frequency": "600",
   "presenceFilter": [
      "person/mood",
      "service/org.openmobilealliance:IM-Session/1.0"
   ]
}}
```

Response:

```
HTTP/1.1 200 OK
```

```
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"presenceListSubscription": {
   "applicationTag": "myApp",
   "callbackReference": {
      "callbackData": "1234",
      "notifyURL": "http://application.example.com/notifications/presenceListNotification"
   },
   "clientCorrelator": "321",
   "duration": "7200",
   "frequency": "600",
   "presenceFilter": [
      "person/mood",
      "service/org.openmobilealliance:IM-Session/1.0"
   ],
   "presenceListId": "myFriends",
   "resourceURL": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-
101/subscriptions/presenceListSubscriptions/myFriends/\n   sub001"
}}
```

D.63 Terminating individual Presence list subscription (section 5.30.6.1)

Request:

```
DELETE /exampleAPI/1/presence/tel%3A%2B1-555-101/subscriptions/presenceListSubscriptions/myFriends/sub001 HTTP/1.1
Host: example.com:80
```

Response:

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

D.64 Notifying Watcher about Presence data updates relating to Presence list (section 5.31.5.1)

Request:

```
POST /notifications/presenceListNotification HTTP/1.1
Host: example.com:80
Content-Type: application/json
Accept: application/json
Content-Length: nnnn

{"presenceListNotification": {
   "callbackData": "1234",
   "link": {
```

```
      "href": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-
100/subscriptions/presenceListSubscriptions/myFriends/sub001",
      "rel": "PresenceListSubscription"
   },
   "presenceList": {"presenceContact": [
      {
         "presence": {"person": {"mood": {"moodValue": "Happy"}}},
         "presentityUserId": "tel:+1-555-101",
         "resourceStatus": "Active"
      },
      {
         "presentityUserId": " tel:+1-555-102",
         "resourceStatus": "Pending"
      }
   ]},
   "presenceListId": "myFriends",
   "resourceStatus": "Active"
}}
```

Response:

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

D.65 Notifying Watcher about termination of Presence list subscription (No resource) (section 5.31.5.2)

Request:

```
POST /notifications/presenceListNotification HTTP/1.1
Host: example.com:80
Content-Type: application/json
Accept: application/json
Content-Length: nnnn

{"presenceListNotification": {
   "callbackData": "1234",
   "presenceListId": "myFriends",
   "resourceStatus": "TerminatedNoResource"
}}
```

Response:

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

D.66 Notifying Watcher about termination of Presence subscription (reason unknown) (section 5.31.5.3)

Request:

```
POST /notifications/presenceListNotification HTTP/1.1
Host: example.com:80
Content-Type: application/json
Accept: application/json
Content-Length: nnnn

{"presenceListNotification": {
   "callbackData": "1234",
   "link": {
      "href": "http://example.com/exampleAPI/1/presence/tel%3A%2B1-555-
101/subscriptions/presenceListSubscriptions/myFriends/sub001",
      "rel": "PresenceListSubscription"
   },
   "presenceListId": "myFriends",
   "resourceStatus": "TerminatedOther"
}}
```

Response:

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

D.67 Notifying Watcher about subscription time out (section 5.31.5.4)

Request:

```
POST /notifications/presenceListNotification HTTP/1.1
Host: example.com:80
Content-Type: application/json
Accept: application/json
Content-Length: nnnn

{"presenceListNotification": {
   "callbackData": "1234",
   "link": {
      "href": "http://example.com/exampleAPI/1/presence/{userId}/subscriptions/presenceListSubscriptions/myFriends/sub001",
      "rel": "PresenceListSubscription"
   },
   "presenceListId": "myFriends",
   "resourceStatus": "TerminatedTimeout"
}}
```

Response:

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

Appendix E. Parlay X operations mapping (Informative)

The table below illustrates the mapping between REST resources/operations and Parlay X equivalent operations.

ParlayREST Resource	ParlayREST Method	ParlayREST Section reference	Parlay X equivalent operation
Presence sources	POST	5.4.5	publish
Individual presence source	PUT	5.5.4	publish
Individual presence source attribute	PUT	5.6.4	publish
Watchers list	GET	5.11.3	getMyWatchers
Individual Watcher	GET	5.12.3	getSubscribedAttributes
Authorization rules	POST	5.13.5	updateAuthorizationRule
Individual authorization rule	PUT	0	updateAuthorizationRule
	DELETE	5.14.6	deleteAuthorizationRule
Presence information by Watcher	GET	5.16.3	getUserPresence
Individual presence attribute by Watcher	GET	5.17.3	getUserPresence
Presence information by Watcher for a member list	GET	5.18.3	getUserPresence
All Watchers subscriptions	POST	5.21.5	startMyWatcherNotification
Individual Watchers subscription	PUT	5.22.4	startMyWatcherNotification
	DELETE	5.22.6	endMyWatchersNotification
Watchers notification	POST	5.23.5	notifyMyWatchers, notifyMyWatchersEnd, NotifyError
Presence subscriptions	POST	5.25.5	startPresenceNotification
Individual Presence subscription	PUT	5.26.4	startPresenceNotification
	DELETE	5.26.6	endPresenceNotification
Presence notification	POST	5.27.5	statusNotified, statusEnd, subscriptionEnded
Presence list subscriptions	POST	5.29.5	startPresenceNotification
Individual presence list subscription	PUT	0	startPresenceNotification
	DELETE	5.30.6	endPresenceNotification
Presence list notification	POST	5.31.5	statusNotified, statusEnd, subscriptionEnded

Table 1: Parlay X operations mapping

Appendix F. Light-weight resources for Presence (Informative)

The following table lists all Presence data structure elements that can be accessed individually as light-weight resources. For each light-weight resource there are listed: 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
Presence source data (5.2.2)	duration	duration	xsd:int	duration
Presence data (5.2.3)	person	person	PersonAttributes	person
	service	service	ServiceAttributes	service/{serviceId}/{version}
	device	device	DeviceAttributes	device/{deviceId}
Person attributes (5.2.4)	activities	activities	Activities	person/activities
	placeType	placeType	PlaceType	person/placeType
	privacy	privacy	Privacy	person/privacy
	sphere	sphere	Sphere	person/sphere
	mood	mood	Mood	person/mood
	placeIs	placeIs	PlaceIs	person/placeIs
	timeOffset	timeOffset	TimeOffset	person/timeOffset
	statusIcon	statusIcon	StatusIcon	person/statusIcon
	class	class	xsd:token	person/class
	noteList	noteList	NoteList	person/noteList
	location	location	Location	person/location
	overridingWillingness	overridingWillingness	OverridingWillingness	person/overridingWillingness
	linkList	linkList	LinkList	person/linkList
	card	card	xsd:anyURI	person/card
	displayName	displayName	xsd:string	person/displayName
	homePage	homePage	xsd:anyURI	person/homePage
	icon	icon	xsd:anyURI	person/icon
	map	map	xsd:anyURI	person/map

	sound	sound	xsd:anyURI	person/sound
	timestamp	timestamp	xsd:dateTime	person/timestamp
	extended	extended	ExtendedList	person/extended
Service attributes (5.2.5)	statusIcon	statusIcon	StatusIcon	service/{serviceId}/{version}/statusIcon
	class	class	xsd:token	service/{serviceId}/{version}/class
	displayName	displayName	xsd:string	service/{serviceId}/{version}/displayName
	homePage	homePage	xsd:anyURI	service/{serviceId}/{version}/homepage
	icon	icon	xsd:anyURI	service/{serviceId}/{version}/icon
	map	map	xsd:anyURI	service/{serviceId}/{version}/map
	sound	sound	xsd:anyURI	service/{serviceId}/{version}/sound
	linkList	linkList	LinkList	service/{serviceId}/{version}/links
	serviceAvailability	serviceAvailability	OpenOrClosed	service/{serviceId}/{version}/serviceAvailability
	serviceWillingness	serviceWillingness	OpenOrClosed	service/{serviceId}/{version}/serviceWillingness
	contact	contact	Contact	service/{serviceId}/{version}/contact
	sessionParticipation	sessionParticipation	OpenOrClosed	service/{serviceId}/{version}/sessionParticipation
	registrationState	registrationState	ActiveOrTerminated	service/{serviceId}/{version}/registrationState
	barringState	barringState	ActiveOrTerminated	service/{serviceId}/{version}/barringState
	sessionAnswerMode	sessionAnswerMode	AutomaticOrManual	service/{serviceId}/{version}/sessionAnswerMode
	devices	devices	DeviceIdentityList	service/{serviceId}/{version}/devices
	timestamp	timestamp	xsd:dateTime	service/{serviceId}/{version}/timestamp
	extended	extended	ExtendedList	service/{serviceId}/{version}/extended

Device attributes (5.2.6)	class	class	xsd:token	device/{deviceId}/class
	location	location	Location	device/{deviceId}/location
	networkAvailability	networkAvailability	NetworkAvailability	device/{deviceId}/networkAvailability
	timestamp	timestamp	xsd:dateTime	device/{deviceId}/timestamp
	extended	extended	ExtendedList	device/{deviceId}/extended
Rule data (5.2.13)	watcherUserId	watcherUserId	xsd:anyURI	watchers/{watcherUserId}
	memberListId	memberListId	xsd:anyURI	memberLists/{memberListId}
	domainName	domainName	xsd:string	domains/{domainName}

Table 2: Light-weight resources for Presence

Note: When appending [ResourceRelPath] string to its heavy-weight resource URL, all variables within curly brackets "{}" such as: serviceId, version, deviceId, watcherUserId, memberListId, and domainName have to be replaced by their real values.