

ETSI GS MEC 013 V3.2.1 (2024-11)



Multi-access Edge Computing (MEC); Location API

Disclaimer

The present document has been produced and approved by the Multi-access Edge Computing (MEC) ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG. It does not necessarily represent the views of the entire ETSI membership.

Reference

RGS/MEC-0013v321LocationApi

Keywords

API, location, MEC, service

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Important notice

The present document can be downloaded from the
[ETSI Search & Browse Standards](#) application.

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format on [ETSI deliver](#) repository.

Users should be aware that the present document may be revised or have its status changed,
this information is available in the [Milestones listing](#).

If you find errors in the present document, please send your comments to
the relevant service listed under [Committee Support Staff](#).

If you find a security vulnerability in the present document, please report it through our
[Coordinated Vulnerability Disclosure \(CVD\)](#) program.

Notice of disclaimer & limitation of liability

The information provided in the present deliverable is directed solely to professionals who have the appropriate degree of experience to understand and interpret its content in accordance with generally accepted engineering or other professional standard and applicable regulations.

No recommendation as to products and services or vendors is made or should be implied.

No representation or warranty is made that this deliverable is technically accurate or sufficient or conforms to any law and/or governmental rule and/or regulation and further, no representation or warranty is made of merchantability or fitness for any particular purpose or against infringement of intellectual property rights.

In no event shall ETSI be held liable for loss of profits or any other incidental or consequential damages.

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

Copyright Notification

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.

The content of the PDF version shall not be modified without the written authorization of ETSI.

The copyright and the foregoing restriction extend to reproduction in all media.

© ETSI 2024.
All rights reserved.

Contents

Intellectual Property Rights	8
Foreword.....	8
Modal verbs terminology.....	8
1 Scope	9
2 References	9
2.1 Normative references	9
2.2 Informative references.....	10
3 Definition of terms, symbols and abbreviations.....	10
3.1 Terms.....	10
3.2 Symbols.....	10
3.3 Abbreviations	10
4 Overview	11
5 Description of the service (informative).....	11
5.1 Introduction	11
5.2 Relation with OMA	12
5.2.1 Void	12
5.2.2 Void	12
5.2.3 General.....	12
5.3 Sequence diagrams	12
5.3.1 Introduction.....	12
5.3.2 UE Location Lookup	12
5.3.3 Void	13
5.3.4 UE Location Subscribe	13
5.3.5 Zone Location Event Subscribe	13
5.3.5A Zone Status Subscribe.....	14
5.3.6 Subscribe Cancellation	15
5.3.6A Zone List Lookup	15
5.3.7 Access Point List Lookup	15
5.3.8 Void	16
5.3.9 UE Distance Lookup.....	16
5.3.10 UE Distance Subscribe	16
5.3.11 UE Area Subscribe.....	17
6 Data Model	18
6.1 Introduction	18
6.2 Resource data types	18
6.2.1 Zonal presence data types	18
6.2.1A Terminal location data types	18
6.2.2 Type: UserInfo.....	18
6.2.3 Type: RelativeLocationInfo	19
6.2.4 Type: MapInfo	19
6.2.5 Type: UserList	20
6.3 Subscription data types.....	20
6.3.1 Void	20
6.3.2 Void	20
6.3.3 Type: NotificationSubscriptionList	20
6.3.4 Type: UserLocationEventSubscription	21
6.3.5 Type: UserLocationPeriodicSubscription	21
6.3.6 Type: ZoneLocationEventSubscription	22
6.3.7 Type: ZoneStatusSubscription	23
6.3.8 Type: UserAreaSubscription.....	24
6.3.9 Type: UserDistanceSubscription	24
6.4 Notifications data types	26
6.4.1 Void	26

6.4.2	Void	26
6.4.3	Type: TestNotification	26
6.4.4	Type: UserLocationEventNotification	26
6.4.5	Type: UserLocationPeriodicNotification	27
6.4.6	Type: ZoneLocationEventNotification	27
6.4.7	Type: ZoneStatusNotification	27
6.4.8	Type: UserAreaNotification	28
6.4.9	Type: UserDistanceNotification	28
6.5	Referenced structured data types	29
6.5.1	Introduction	29
6.5.2	Type: TimeStamp	29
6.5.3	Type: LocationInfo	29
6.5.4	Type: WebsockNotifConfig	30
6.5.5	Type: UserEventPara	30
6.5.6	Type: ReportingCtrl	31
6.5.7	Type: AreaInfo	31
6.5.8	Type: Point	31
6.6	Mapping of Identifiers	32
6.6.1	accessPointId	32
6.6.2	userId	32
6.7	Referenced simple data types and enumerations	32
6.7.1	Introduction	32
6.7.2	Type: LinkType	32
6.7.3	Enumeration: LocationEventType	32
6.7.4	Enumeration: NotificationResult	32
6.7.5	Enumeration: ConnectionType	33
7	API definition	33
7.1	Introduction	33
7.2	Global definitions and resource structure	33
7.3	Void	35
7.4	Resource: users	35
7.4.1	Description	35
7.4.2	Resource definition	35
7.4.3	Resource methods	35
7.4.3.1	GET	35
7.4.3.2	PUT	36
7.4.3.3	PATCH	36
7.4.3.4	POST	36
7.4.3.5	DELETE	36
7.5	Resource: user_subscriptions	37
7.5.1	Description	37
7.5.2	Resource definition	37
7.5.3	Resource methods	37
7.5.3.1	GET	37
7.5.3.2	PUT	38
7.5.3.3	PATCH	38
7.5.3.4	POST	38
7.5.3.5	DELETE	40
7.6	Resource: individual user_subscription	40
7.6.1	Description	40
7.6.2	Resource definition	40
7.6.3	Resource methods	40
7.6.3.1	GET	40
7.6.3.2	PUT	41
7.6.3.3	PATCH	43
7.6.3.4	POST	43
7.6.3.5	DELETE	43
7.7	Resource: zones	44
7.7.1	Description	44
7.7.2	Resource definition	44
7.7.3	Resource methods	44

7.7.3.1	GET	44
7.7.3.2	PUT	45
7.7.3.3	PATCH	45
7.7.3.4	POST	45
7.7.3.5	DELETE	45
7.8	Resource: individual zone	46
7.8.1	Description	46
7.8.2	Resource definition	46
7.8.3	Resource methods	46
7.8.3.1	GET	46
7.8.3.2	PUT	47
7.8.3.3	PATCH	47
7.8.3.4	POST	47
7.8.3.5	DELETE	47
7.9	Resource: accessPoints	47
7.9.1	Description	47
7.9.2	Resource definition	47
7.9.3	Resource methods	48
7.9.3.1	GET	48
7.9.3.2	PUT	48
7.9.3.3	PATCH	49
7.9.3.4	POST	49
7.9.3.5	DELETE	49
7.10	Resource: individual accessPoint	49
7.10.1	Description	49
7.10.2	Resource definition	49
7.10.3	Resource methods	49
7.10.3.1	GET	49
7.10.3.2	PUT	50
7.10.3.3	PATCH	50
7.10.3.4	POST	50
7.10.3.5	DELETE	50
7.11	Resource: zone_subscriptions	51
7.11.1	Description	51
7.11.2	Resource definition	51
7.11.3	Resource methods	51
7.11.3.1	GET	51
7.11.3.2	PUT	52
7.11.3.3	PATCH	52
7.11.3.4	POST	52
7.11.3.5	DELETE	53
7.12	Resource: individual zone_subscription	54
7.12.1	Description	54
7.12.2	Resource definition	54
7.12.3	Resource methods	54
7.12.3.1	GET	54
7.12.3.2	PUT	55
7.12.3.3	PATCH	56
7.12.3.4	POST	56
7.12.3.5	DELETE	57
7.13	Resource: distance	57
7.13.1	Description	57
7.13.2	Resource definition	57
7.13.3	Resource methods	58
7.13.3.1	GET	58
7.13.3.2	PUT	59
7.13.3.3	PATCH	59
7.13.3.4	POST	59
7.13.3.5	DELETE	59
7.14	Resource: distance_subscriptions	59
7.14.1	Description	59
7.14.2	Resource definition	59

7.14.3	Resource methods.....	59
7.14.3.1	GET.....	59
7.14.3.2	PUT.....	60
7.14.3.3	PATCH.....	60
7.14.3.4	POST.....	60
7.14.3.5	DELETE.....	62
7.15	Resource: individual distance_subscription.....	62
7.15.1	Description.....	62
7.15.2	Resource definition.....	62
7.15.3	Resource methods.....	62
7.15.3.1	GET.....	62
7.15.3.2	PUT.....	63
7.15.3.3	PATCH.....	65
7.15.3.4	POST.....	65
7.15.3.5	DELETE.....	65
7.16	Resource: area_subscriptions.....	66
7.16.1	Description.....	66
7.16.2	Resource definition.....	66
7.16.3	Resource methods.....	66
7.16.3.1	GET.....	66
7.16.3.2	PUT.....	67
7.16.3.3	PATCH.....	67
7.16.3.4	POST.....	67
7.16.3.5	DELETE.....	69
7.17	Resource: individual area_subscription.....	69
7.17.1	Description.....	69
7.17.2	Resource definition.....	69
7.17.3	Resource methods.....	69
7.17.3.1	GET.....	69
7.17.3.2	PUT.....	70
7.17.3.3	PATCH.....	72
7.17.3.4	POST.....	72
7.17.3.5	DELETE.....	72
Annex A (informative):	Complementary material for API utilization.....	74
Annex B (informative):	Examples for the usage of Location API.....	75
B.1	Examples for UE Location Lookup.....	75
B.1.1	Example: Retrieve UE location by the address of a user.....	75
B.1.2	Example: Retrieve UE Location for the users in a zone.....	75
B.2	Examples for UE Location Subscribe.....	76
B.2.1	Example: create a UE Location event subscription.....	76
B.2.2	Example: client notification about UE entering an area.....	77
B.2.3	Example: create a UE Location periodic subscription.....	77
B.2.4	Example: client notification about UE location in a fixed interval.....	78
B.3	Examples for Zone List Lookup and Access Point List Lookup.....	78
B.3.1	Example: Retrieve zone list.....	78
B.3.2	Example: Retrieve Access Point List.....	79
B.4	Examples for Zone Location Event Subscribe and Zone Status Subscribe.....	80
B.4.1	Example: create a zone location subscription.....	80
B.4.2	Example: client notification about UE entering the zone.....	80
B.4.3	Example: create a zone status subscription.....	81
B.4.4	Example: client notification about UE number reaching the threshold.....	81
B.5	Examples for UE Distance Lookup.....	82
B.5.1	Example: retrieve distance between two UEs.....	82
B.6	Examples for UE Distance Subscribe.....	82
B.6.1	Example: create a UE Distance event subscription.....	82
B.6.2	Example: client notification about all monitored devices within the specified distance.....	83

B.7	Examples for UE Area Subscribe.....	84
B.7.1	Example: Create a UE Area subscription	84
B.7.2	Example: Client notification about UE Area change	85
History	86

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: "*Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards*", which is available from the ETSI Secretariat. Latest updates are available on the [ETSI IPR online database](#).

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™**, **LTE™** and **5G™** logo are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM®** and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

This Group Specification (GS) has been produced by ETSI Industry Specification Group (ISG) Multi-access Edge Computing (MEC).

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the [ETSI Drafting Rules](#) (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

1 Scope

The present document focuses on the MEC Location Service. It describes the related application policy information including authorization and access control, information flows, required information and service aggregation patterns. The present document specifies the necessary API with the data model and data format.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at [ETSI docbox](#).

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

- [1] Void.
- [2] [ETSI GS MEC 002](#): "Multi-access Edge Computing (MEC); Use Cases and Requirements".
- [3] [ETSI GS MEC 003](#): "Multi-access Edge Computing (MEC); Framework and Reference Architecture".
- [4] [ETSI GS MEC 009](#): "Multi-access Edge Computing (MEC); General principles, patterns and common aspects of MEC Service APIs".
- [5] [OMA-TS-REST-NetAPI-ZonalPresence-V1_0-20160308-C](#): "RESTful Network API for Zonal Presence".
- [6] [OMA-TS-REST-NetAPI-TerminalLocation-V1_0_1-20151029-A](#): "RESTful Network API for Terminal Location".
- [7] [OMA-TS-REST-NetAPI-ACR-V1_0-20151201-C](#): "RESTful Network API for Anonymous Customer Reference Management".
- [8] [ETSI TS 129 171](#): "Digital cellular telecommunications system (Phase 2+) (GSM); Universal Mobile Telecommunications System (UMTS); LTE; Location Services (LCS); LCS Application Protocol (LCS-AP) between the Mobile Management Entity (MME) and Evolved Serving Mobile Location Centre (E-SMLC); SLs interface (3GPP TS 29.171)".
- [9] Void.
- [10] [IETF RFC 5246](#): "The Transport Layer Security (TLS) Protocol Version 1.2".
- [11] [IETF RFC 6749](#): "The OAuth 2.0 Authorization Framework".
- [12] [IETF RFC 6750](#): "The OAuth 2.0 Authorization Framework: Bearer Token Usage".
- [13] [National Geospatial-Intelligence Agency \(NGA\) Standardization Document](#): "Department of Defense, World Geodetic System 1984, Its Definition and Relationships with Local Geodetic Systems", NGA.STND.0036_1.0.0_WGS84, 2014-07-08, Version 1.0.0.
- [14] [ETSI TS 123 032](#): "Digital cellular telecommunications system (Phase 2+) (GSM); Universal Mobile Telecommunications System (UMTS); Universal Geographical Area Description (GAD) (3GPP TS 23.032)".

- [15] [IETF RFC 6225](#): "Dynamic Host Configuration Protocol Options for Coordinate-Based Location Configuration Information".
- [16] [IETF RFC 8446](#): "The Transport Layer Security (TLS) Protocol Version 1.3".
- [17] [ETSI TS 129 572](#): "5G System; Location Management Services; Stage 3 (3GPP TS 29.572)".

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

- [i.1] SCF 084.07.01: "Small cell zone services - RESTful bindings".
- [i.2] SCF 152.07.01: "Small cell services API".
- [i.3] [OpenAPI™ Specification](#).
- [i.4] [ETSI GS MEC 001](#): "Multi-access Edge Computing (MEC); Terminology".

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the terms given in ETSI GS MEC 001 [i.4] and the following apply:

Anonymous Customer Reference (ACR): Uniform Resource Identifier (URI) scheme describing an anonymous reference that can be mapped to a resource or user/user group

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the abbreviations given in ETSI GS MEC 001 [i.4] and the following apply:

ACR	Anonymous Customer Reference
API	Application Programming Interface
E-SMLC	Evolved Serving Mobile Location Centre
ID	IDentifier
LS	Location Service
OMA	Open Mobile Alliance
REST	REpresentational State Transfer
SCF	Small Cell Forum
UE	User Equipment
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
WGS	World Geodetic System

4 Overview

The present document specifies the Location Service API to support the requirements defined for Multi-access Edge Computing in ETSI GS MEC 002 [2]. The Location Service (LS) defined in the present document leverages the Zonal Presence service described in SCF 084.07.01 [i.1] and in SCF 152.07.01 [i.2]. The present document references some data types defined in the Open Mobile Alliance (OMA) specifications "RESTful Network API for Zonal Presence" [5] and "RESTful Network API for Terminal Location" [6].

In addition, the present document contains application instructions on how the Anonymous Customer Reference (ACR) defined in [7] can be applied to fulfil the ETSI MEC requirement [Location-04] on addressing user categories defined in ETSI GS MEC 002 [2], and application instructions on how the 3GPP Cell Identifiers defined in ETSI TS 129 171 [8] can be mapped to the Access Point identifier of the OMA API [5].

The causes of the dramatic change from V2 to V3 are as follows:

- 1) To support the Websocket feature, several new data types and resources need to be defined instead of being referenced from the OMA specifications, e.g. UserLocationEventSubscription, UserLocationPeriodicSubscription, etc.
- 2) Civic address and relative location are added to enhance the location services.
- 3) For better understanding, some modifications are implemented, including the combination of some similar procedures and deletion of redundant procedures.

5 Description of the service (informative)

5.1 Introduction

Location Service is a service to provide the location related information to the MEC platform or authorized applications. With location related information, the MEC platform or applications perform the active device location tracking, location-based service recommendation, etc., see ETSI GS MEC 002 [2]. The Location Service is registered and discovered over the Mp1 reference point defined in ETSI GS MEC 003 [3].

The Location Service supports the location retrieval mechanism, i.e. the location is reported only once for each location information request.

The Location Service supports the location subscribe mechanism, i.e. the location is able to be reported multiple times for each location request, periodically or based on specific events, such as location change.

The Location Service supports the anonymous location report, i.e. without the related UE ID information, e.g. for the statistics collection.

The Location Service supports the following location information:

- the location information of specific UEs currently served by the radio node(s) associated with the MEC host;
- the location information of all UEs currently served by the radio node(s) associated with the MEC host;
- the distance between specified UEs currently served by the radio node(s) associated with the MEC host;
- the distance between a specified location and a UE currently served by the radio node(s) associated with the MEC host;
- the location information of a certain category of UEs currently served by the radio node(s) associated with the MEC host;
- a list of UEs in a particular location area;
- the specific UEs which move in or out of a particular location area;
- information about the location of all radio nodes currently associated with the MEC host;

- subscriptions to location information are also offered, including periodic location information updates, updates on changes in distance and location updates relating to UEs in a particular location area.

The Location Service supports both geolocation, such as a geographical coordinates, and logical location, such as a Cell ID.

5.2 Relation with OMA

5.2.1 Void

5.2.2 Void

5.2.3 General

The Location Service before version 3.1.1 references the APIs and the data types defined in [5] and [6] with slight modifications on the URI of the service. Beginning with version 3.1.1, the LS defines its data types and APIs in support of other location information (e.g. civic address, relative location) and the Websocket feature of the subscription. However, the present document still references some data types to avoid duplicate work. See clause 6.2.1 and clause 6.2.1A for detailed information.

5.3 Sequence diagrams

5.3.1 Introduction

Clauses 5.3.2 to 5.3.11 describe how the Service Consumers interact with the Location Service over LS API to obtain location information of a UE, a group of UEs or the radio nodes currently associated with the MEC host. The sequence diagrams that are relevant for the Location Service are presented.

The Service Consumers communicate with the Location Service over the LS API. Both the MEC application and MEC platform can be Service Consumers. Location information can be provided by both MEC platform and MEC application.

The Location Service API supports both queries and subscriptions (pub/sub mechanism) that can be used over RESTful API as per the mechanism in OMA APIs [5] and [6].

5.3.2 UE Location Lookup

The UE Location Lookup is the procedure for applications acquiring the current location information of a specific UE or a group of UEs.

The UE Location Lookup procedure is illustrated in figure 5.3.2-1.

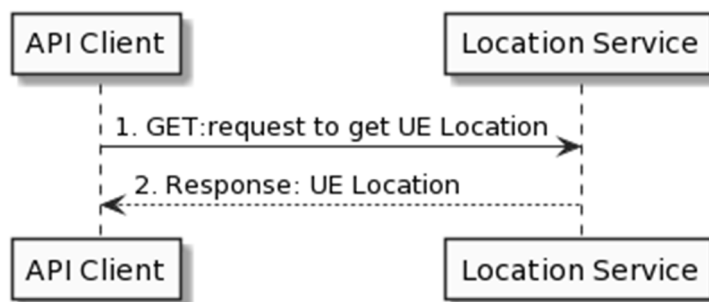


Figure 5.3.2-1: Flow of UE Location Lookup

- 1) The MEC application looks up UE location by sending a request to the resource representing the UE location, which includes location area information. The request may optionally include one or more query parameters specifying the sub-region of interest, the Access Point identifier and requirements on reporting timeliness and accuracy.
- 2) The Location Service returns a response with a message body including the UE location of the UE(s) according to the query parameters, if the UE location lookup is accepted.

5.3.3 Void

5.3.4 UE Location Subscribe

The UE Location Subscribe is the procedure for applications to request to receive notifications about some events happened, e.g. location information changes of a specific UE or a group of UEs or timer expiration, which for instance may help the applications to regularly track the UE(s) location. In this procedure, the Location Service will continue to report the subscribed information until the subscription is cancelled, or an optional specified time limit.

The UE Location Subscribe procedure is illustrated in figure 5.3.4-1.

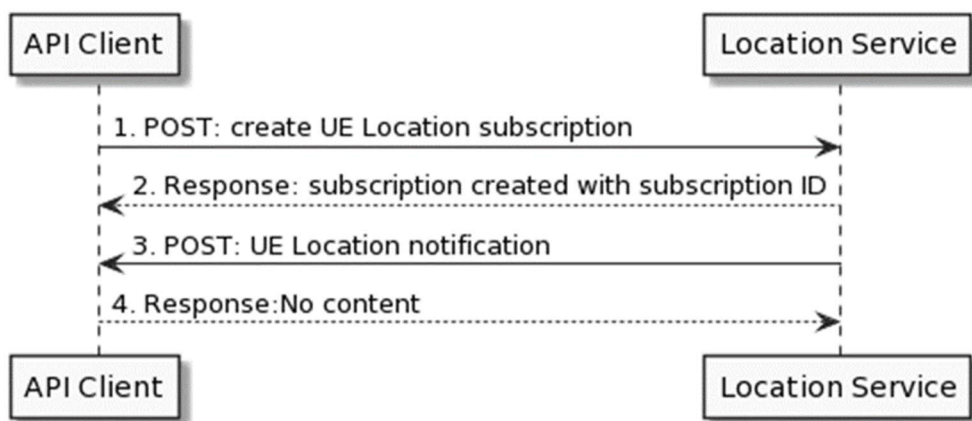


Figure 5.3.4-1: Flow of UE Location Subscribe

- 1) The MEC application subscribes to UE location notification by requesting the creation of a resource containing the subscription details, which includes UE(s) identifier, e.g. UE IP address, and a callbackURL for receiving the UE location.
- 2) The Location Service returns a response with resource URI containing the subscriptionId.
- 3) The Location Service reports the up-to-date subscribed information to the MEC application by sending a message with the message body containing the UE Location notification to the callbackURL, which includes location information.
- 4) The MEC application returns a response with the code 204.

5.3.5 Zone Location Event Subscribe

The Zone Location Event Subscribe is the procedure for API clients to receive notifications of UE location events related with a zone, e.g. UE entering the zone, UE leaving the zone. In this procedure, the Location Service will continue to report the subscribed information until the subscription is cancelled.

The Zone Location Event Subscribe procedure is illustrated in figure 5.3.5-1.

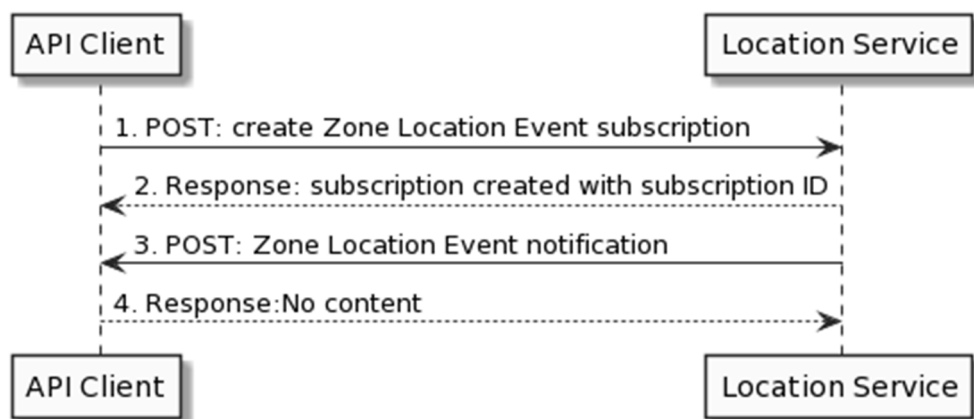


Figure 5.3.5-1: Flow of Zone Location Event Subscribe

- 1) The API Client (e.g. MEC application) subscribes to the Zone Location Event notification for a zone by requesting the creation of a resource containing all subscription details, which includes zone Id and a callbackURL for receiving the notifications.
- 2) The Location Service returns a response with resource URI containing the subscriptionId.
- 3) The Location Service reports the subscribed information to the API Client by sending a message with the request body containing the UE location event related with the zone to the callbackURL.
- 4) The API Client returns a response with the code 204.

5.3.5A Zone Status Subscribe

The Zone Status Subscribe is the procedure for applications to receive notifications of the number of UEs reaching threshold or the status of access point changing. In this procedure, the Location Service will continue to report the subscribed information until the subscription is cancelled.

The Zone Status Subscribe procedure is illustrated in figure 5.3.5A-1.

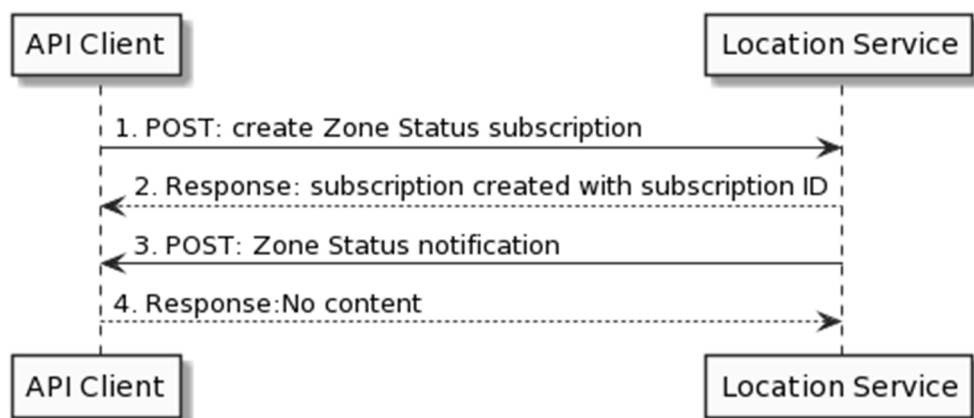


Figure 5.3.5A-1: Flow of Zone Status Subscribe

- 1) The API Client (e.g. MEC application) subscribes to the Zone Status notification for a particular area by requesting the creation of a resource containing all subscription details, which includes zone Id and a callbackURL for receiving the notifications.
- 2) The Location Service returns a response with resource URI containing the subscriptionId.
- 3) The Location Service reports the subscribed information to the API Client by sending a message with the request body containing the zone status notification to the callbackURL.

- 4) The API Client returns a response with the code 204.

5.3.6 Subscribe Cancellation

The Subscribe Cancellation is the procedure for applications to cancel the subscription, with which the Location Service stop reporting the subscribed information to the application.

The Subscribe Cancellation procedure is illustrated in figure 5.3.6-1.

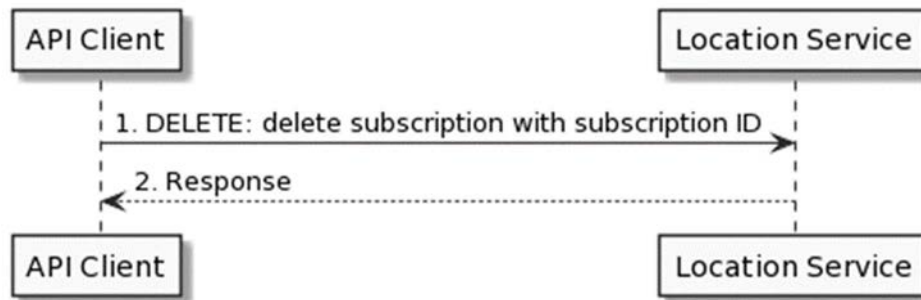


Figure 5.3.6-1: Flow of Location Subscribe Cancellation

- 1) The MEC application unsubscribes to notifications by sending a request to delete the resource URI containing the subscriptionId.
- 2) The Location Service returns a successful response if the subscription cancellation is accepted.

5.3.6A Zone List Lookup

The Zone List Lookup procedure enables an API client to make an enquiry about all the zones currently associated with the MEC host. The Zone List Lookup procedure is illustrated in figure 5.3.6A-1.

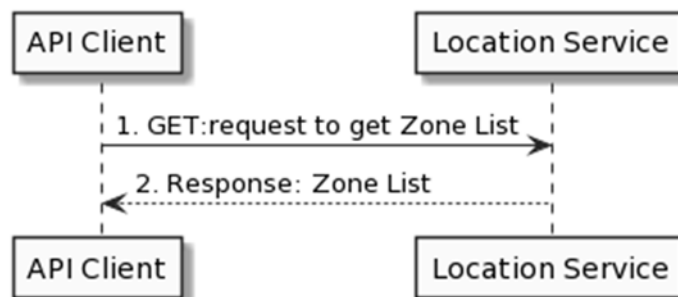


Figure 5.3.6A-1: Zone List Lookup

- 1) The API Client (e.g. MEC application) makes an enquiry about the zones currently associated with the MEC host by sending a request to the resource representing zones information.
- 2) The Location Service returns a response with message body including the list of zones currently associated with the MEC host.

5.3.7 Access Point List Lookup

The Access Point List Lookup procedure enables an API client to make an information enquiry about the access points currently associated with the MEC host. The Access Point List Lookup procedure is illustrated in figure 5.3.7-1.



Figure 5.3.7-1: Access Point List Lookup

- 1) The API Client (e.g. MEC application) makes an enquiry about the access points currently associated with the MEC host by sending a request to the resource representing access point information, e.g. E-CGI as per ETSI TS 129 171 [8].
- 2) The Location Service returns a response with message body including the list of access points currently associated with the MEC host and the information of each access point.

5.3.8 Void

5.3.9 UE Distance Lookup

The UE Distance Lookup is the procedure for applications acquiring the current distance of a specific UE to a geographical location, or another UE.

The UE Distance Lookup procedure is illustrated in figure 5.3.9-1.

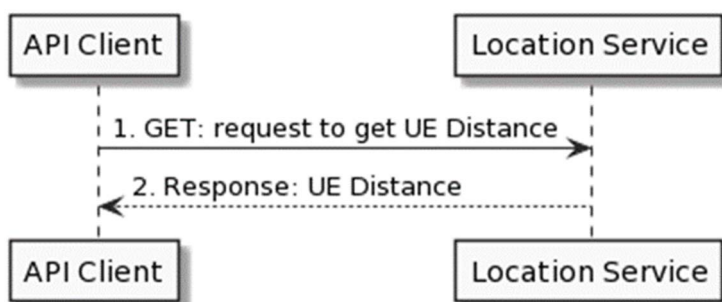


Figure 5.3.9-1: Flow of UE Distance Lookup

- 1) The MEC application looks up an UE distance by sending a request to the resource representing the UE distance, which includes the two UE identities, e.g. UE IP address, or a single UE identifier plus the coordinates of the location to measure from.
- 2) The Location Service returns a response with a message body including the distance information if the UE location lookup is accepted.

5.3.10 UE Distance Subscribe

The UE Distance Subscribe is the procedure for applications acquiring up-to-data distance of a specific UE to a geographical location, or another UE. In this procedure, the Location Service will continue to report the subscribed information until the subscription is cancelled.

The UE Distance Subscribe procedure is illustrated in figure 5.3.10-1.

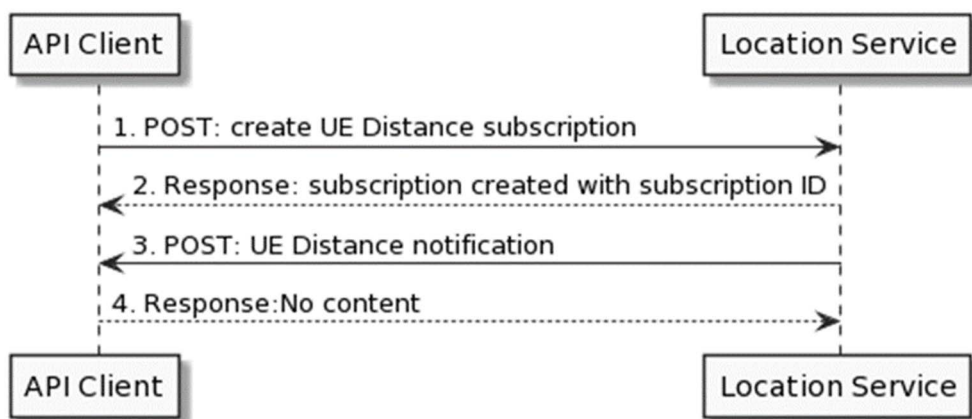


Figure 5.3.10-1: Flow of UE Distance Subscribe

- 1) The MEC application subscribes to UE distance notification by requesting the creation of a resource containing the subscription details, with a callbackURL for receiving the UE distance. It also includes the UE identities, e.g. UE IP address, between which to calculate the distances, the required accuracy and the minimum interval between notifications.
- 2) The Location Service returns a response with resource URI containing the subscriptionId.
- 3) The Location Service reports the up-to-date subscribed information to the MEC application by sending a message with the message body containing the UE Distance notification to the callbackURL, which includes distance information.
- 4) The MEC application returns a response with the code 204.

5.3.11 UE Area Subscribe

The UE Area Subscribe is the procedure for applications acquiring UE movement notifications in relation to a geographic area. In this procedure, the Location Service will continue to report the subscribed information until the subscription is cancelled.

The UE Area Subscribe procedure is illustrated in figure 5.3.11-1.

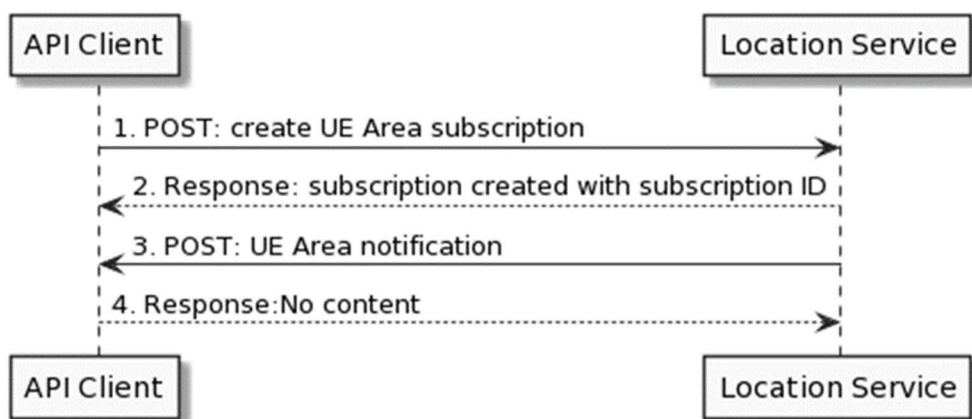


Figure 5.3.11-1: Flow of UE Area Subscribe

- 1) The MEC application subscribes to UE Area notifications by requesting the creation of a resource containing the subscription details, which includes the UE identities, e.g. UE IP address, and a callbackURL for receiving the UE Area notification. Also included is the area definition, the required accuracy, the minimum interval between notifications and whether notifications should be generated on entering, or leaving the area, or both.
- 2) The Location Service returns a response with resource URI containing the subscriptionId.

- 3) The Location Service reports the up-to-data subscribed information to the MEC application by sending a message with the message body containing the UE Area notification to the callbackURL, which includes area information.
- 4) The MEC application returns a response with the code 204.

6 Data Model

6.1 Introduction

Data types from the APIs defined in [5] and [6] are reused. The following application instructions describe the mapping between the terminology used in the OMA specifications and the terminologies used in ETSI MEC and ETSI TS 129 171 [8].

The elements specified in OMA APIs' data models in [5] and [6] shall be represented as a JSON object. Moreover, the elements therein with upperbound cardinality greater than 1, e.g. "0.. N", "1.. N", etc., shall be represented as a JSON array.

6.2 Resource data types

6.2.1 Zonal presence data types

The data structures to be used in resource representations for the Location Service API consist in the following data types defined in [5], clause 5:

- **ZoneList**
A type containing list of zones' information;
- **ZoneInfo**
A type containing zone information;
- **AccessPointList**
A type containing list of access points' information;
- **AccessPointInfo**
A type containing access point information; the element **ConnectionType** is redefined in clause 6.7.5.

6.2.1A Terminal location data types

In addition, further data types as defined in [6], clause 5, are offered:

- **TerminalLocation**
A type containing device address, retrieval status and location information;
- **TerminalDistance**
A type containing information about the distance from a terminal to a location or between two terminals.

6.2.2 Type: UserInfo

This type represents the information related to a user attached to an access point associated to the MEC host, such access point is in scope of the Location Service instance.

The **UserInfo** data type extends the **UserInfo** data type defined in [5], and its elements shall follow the notations provided in table 6.2.2-1.

Table 6.2.2-1: Definition of type UserInfo

Attribute name	Data type	Cardinality	Description
address	AnyURI	1	Address of user (e.g. 'sip' URI, 'tel' URI, 'acr' URI) currently on the access point, see note 1.
AccessPointId	String	1	The identity of the access point the user is currently on, see note 1.
zoneId	String	1	The identity of the zone the user is currently within, see note 1.
resourceURL	AnyURI	0..1	Self-referring URL, see note 1. This shall be only included in the procedure of UE Location Lookup.
timeStamp	TimeStamp	1	Date and time that location was collected.
locationInfo	LocationInfo	0..1	The geographical coordinates where the user is.
civicInfo	CivicAddress	0..1	Contextual information of a user location (e.g. aisle, floor, room number, etc.). See note 3.
ancillaryInfo	String	0..1	Reserved for future use.
relativeLocationInfo	RelativeLocationInfo	0..1	The relative location in a reference system.
NOTE 1: As specified in [5], clause 5.2.2.7.			
NOTE 2: Void.			
NOTE 3: As specified in [17], clause 6.1.6.2.14.			

The address sub-element supports the representation of the UE's IP address, as specified in clause 6.6.2.

6.2.3 Type: RelativeLocationInfo

This type represents the relative location in a reference system that is a Cartesian coordinate system and described by a map.

Table 6.2.3-1: Definition of type RelativeLocationInfo

Attribute name	Data type	Cardinality	Description
mapInfo	MapInfo	1	Indicates a map corresponding to a location area.
X	Float	1	Indicates the value (in the unit of meters) on x-axis of the relative location in the Cartesian system. Positive value represents easting from origin.
Y	Float	1	Indicates the value (in the unit of meters) on y-axis of the relative location in the Cartesian system. Positive value represents northing from origin.
Z	Float	0..1	Indicates the value (in the unit of meters) on z-axis of the relative location in the Cartesian system for a 3D-Point. Positive value represents height above origin.

6.2.4 Type: MapInfo

This type represents a map that may incorporate an origin. For the sake of security, the location of map origin could be configured to the service consumer instead of being transported explicitly. The ancillaryMapInfo may help the service consumer do coordinates conversion, e.g. converting Cartesian coordinate to Universal Transverse Mercator (UTM) coordinate.

Table 6.2.4-1: Definition of type MapInfo

Attribute name	Data type	Cardinality	Description
mapId	String	1	Indicates the ID of the map.
origin	Structure(inlined)	0..1	Indicates the location of the map origin in the local Cartesian coordinate system.
>latitude	Float	1	Location latitude, expressed in the range -90° to +90°.
>longitude	Float	1	Location longitude, expressed in the range -180° to +180°.
>altitude	Float	0..1	Location altitude relative to the WGS84 ellipsoid surface.
ancillaryMapInfo	Not specified	0..1	Ancillary map information may be used to convert coordinates between different coordinate systems.

6.2.5 Type: UserList

This type represents a list of UserInfo.

Table 6.2.5-1: Definition of type UserList

Attribute name	Data type	Cardinality	Description
user	Array(UserInfo)	0..N	Collection of the user location information list.
resourceURL	Uri	0..1	Self referring URL. This shall be only included in the procedure of UE Location Lookup.

6.3 Subscription data types

6.3.1 Void

6.3.2 Void

6.3.3 Type: NotificationSubscriptionList

This type contains a list of subscriptions.

Table 6.3.3-1: Definition of type NotificationSubscriptionList

Attribute name	Data type	Cardinality	Description
subscription	Structure (inlined)	0..N	
>href	Uri	1	The URI referring to the subscription.
>subscriptionType	String	1	Type of the subscription. The string shall be set according to the "subscriptionType" attribute of the associated subscription data type defined in clauses 6.3.4, 6.3.5, 6.3.6, 6.3.7, 6.3.8 and 6.3.9: <ul style="list-style-type: none"> • "UserLocationEventSubscription" • "UserLocationPeriodicSubscription" • "ZoneLocationEventSubscription" • "ZoneStatusSubscription" • "UserAreaSubscription" • "UserDistanceSubscription"
resourceURL	LinkType	1	Self-referring URL.
NOTE 1: Void.			
NOTE 2: Void.			

6.3.4 Type: UserLocationEventSubscription

This type represents a subscription to the notifications from location server about events related with user location.

The attributes of the UserLocationEventSubscription shall follow the indications provided in table 6.3.4-1.

Table 6.3.4-1: Definition of type UserLocationEventSubscription

Attribute name	Data type	Cardinality	Description
subscriptionType	String	1	Shall be set to "UserLocationEventSubscription".
clientCorrelator	String	0..1	A correlator that the client can use to tag this particular resource representation during a request to create a resource on the server. See note 2.
callbackReference	Uri	0..1	URI exposed by the client on which to receive notifications via HTTP. See note 1.
requestTestNotification	Boolean	0..1	Set to TRUE by the service consumer to request a test notification via HTTP on the callbackReference URI, as specified in ETSI GS MEC 009 [4], clause 6.12a.
websocketNotifConfig	WebsocketNotifConfig	0..1	Provides details to negotiate and signal the use of a WebSocket connection between the location server and the service consumer for notifications. See note 1.
_links	Structure (inlined)	0..1	Hyperlink related to the resource. This shall be only included in the HTTP responses and in HTTP PUT requests.
>self	LinkType	1	Self-referring URI. The URI shall be unique within the UE Location Subscribe as it acts as an ID for the subscription.
address	Uri	1	Address of user (e.g. 'sip' URI, 'tel' URI, 'acr' URI) to monitor.
userEventPara	UserEventPara	0..1	Requirements for user event reporting.
locationEventCriteria	Array(LocationEvent Type)	0..N	List of user event values to generate notifications for (these apply to address specified).
expiryDeadline	TimeStamp	0..1	The expiration time of the subscription determined by the UE Location Subscribe Service.
NOTE 1: At least one of callbackReference and websocketNotifConfig shall be provided by the service consumer. If both are provided, it is up to location server to select an alternative and return only that alternative in the response, as specified in ETSI GS MEC 009 [4], clause 6.12a.			
NOTE 2: This allows the client to recover from communication failures during resource creation and therefore avoids duplicate subscription creation in such situations. In case the element is present, the server shall not alter its value, and shall provide it as part of the representation of this resource. In case the element is not present, the server shall not generate it.			

6.3.5 Type: UserLocationPeriodicSubscription

This type represents a subscription to the notifications from location server about events triggered by timer expiration.

The attributes of the UserLocationPeriodicSubscription shall follow the indications provided in table 6.3.5-1.

Table 6.3.5-1: Definition of type UserLocationPeriodicSubscription

Attribute name	Data type	Cardinality	Description
subscriptionType	String	1	Shall be set to "UserLocationPeriodicSubscription".
clientCorrelator	String	0..1	A correlator that the client can use to tag this particular resource representation during a request to create a resource on the server. See note 2.
callbackReference	Uri	0..1	URI exposed by the client on which to receive notifications via HTTP. See note 1.
requestTestNotification	Boolean	0..1	Set to TRUE by the service consumer to request a test notification via HTTP on the callbackReference URI, as specified in ETSI GS MEC 009 [4], clause 6.12a.
websocketNotifConfig	WebsocketNotifConfig	0..1	Provides details to negotiate and signal the use of a WebSocket connection between the location server and the service consumer for notifications. See note 1.

Attribute name	Data type	Cardinality	Description
_links	Structure (inlined)	0..1	Hyperlink related to the resource. This shall be only included in the HTTP responses and in HTTP PUT requests.
>self	LinkType	1	Self-referring URI. The URI shall be unique within the UE Location Subscribe as it acts as an ID for the subscription.
address	Uri	1	Address of user (e.g. 'sip' URI, 'tel' URI, 'acr' URI) to monitor.
periodicEventInfo	PeriodicEventInfo	1	Information for periodic event reporting. See note 3.
expiryDeadline	TimeStamp	0..1	The expiration time of the subscription determined by the UE Location Subscribe Service.
NOTE 1: At least one of callbackReference and websocketNotifConfig shall be provided by the service consumer. If both are provided, it is up to location server to select an alternative and return only that alternative in the response, as specified in ETSI GS MEC 009 [4], clause 6.12a.			
NOTE 2: This allows the client to recover from communication failures during resource creation and therefore avoids duplicate subscription creation in such situations. In case the element is present, the server shall not alter its value, and shall provide it as part of the representation of this resource. In case the element is not present, the server shall not generate it.			
NOTE 3: As specified in [17], clause 6.1.6.2.24.			

6.3.6 Type: ZoneLocationEventSubscription

This type represents a subscription to the notifications from location server about user location events happened in the zone.

The attributes of the ZoneLocationEventSubscription shall follow the indications provided in table 6.3.6-1.

Table 6.3.6-1: Definition of type ZoneLocationEventSubscription

Attribute name	Data type	Cardinality	Description
subscriptionType	String	1	Shall be set to "ZoneLocationEventSubscription".
clientCorrelator	String	0..1	A correlator that the client can use to tag this particular resource representation during a request to create a resource on the server. See note 2.
callbackReference	Uri	0..1	URI exposed by the client on which to receive notifications via HTTP. See note 1.
requestTestNotification	Boolean	0..1	Set to TRUE by the service consumer to request a test notification via HTTP on the callbackReference URI, as specified in ETSI GS MEC 009 [4], clause 6.12a.
websocketNotifConfig	WebsocketNotifConfig	0..1	Provides details to negotiate and signal the use of a WebSocket connection between the location server and the service consumer for notifications. See note 1.
_links	Structure (inlined)	0..1	Hyperlink related to the resource. This shall be only included in the HTTP responses and in HTTP PUT requests.
>self	LinkType	1	Self-referring URI. The URI shall be unique within the UE Location Subscribe as it acts as an ID for the subscription.
zoneId	String	1	Identifier of zone (e.g. zone001) to monitor.
addressList	Array(Uri)	0..N	List of the users to be monitored. If not present, all the users need to be monitored.
locationEventCriteria	Array(LocationEventType)	0..N	List of user event values to generate notifications for.
reportingCtrl	ReportingCtrl	0..1	Provides parameters that ctrl the reporting.
expiryDeadline	TimeStamp	0..1	The expiration time of the subscription determined by the Zone Location Event Service.

Attribute name	Data type	Cardinality	Description
NOTE 1:	At least one of callbackReference and websockNotifConfig shall be provided by the service consumer. If both are provided, it is up to location server to select an alternative and return only that alternative in the response, as specified in ETSI GS MEC 009 [4], clause 6.12a.		
NOTE 2:	This allows the client to recover from communication failures during resource creation and therefore avoids duplicate subscription creation in such situations. In case the element is present, the server shall not alter its value, and shall provide it as part of the representation of this resource. In case the element is not present, the server shall not generate it.		

6.3.7 Type: ZoneStatusSubscription

This type represents a subscription to the notifications from location server about the number of users reaching the threshold or the status of access points changing.

The attributes of the ZoneStatusSubscription shall follow the indications provided in table 6.3.7-1.

Table 6.3.7-1: Definition of type ZoneStatusSubscription

Attribute name	Data type	Cardinality	Description
subscriptionType	String	1	Shall be set to "ZoneStatusSubscription".
clientCorrelator	String	0..1	A correlator that the client can use to tag this particular resource representation during a request to create a resource on the server. See note 2.
callbackReference	Uri	0..1	URI exposed by the client on which to receive notifications via HTTP. See note 1.
requestTestNotification	Boolean	0..1	Set to TRUE by the service consumer to request a test notification via HTTP on the callbackReference URI, as specified in ETSI GS MEC 009 [4], clause 6.12a.
websockNotifConfig	WebsockNotifConfig	0..1	Provides details to negotiate and signal the use of a WebSocket connection between the location server and the service consumer for notifications. See note 1.
_links	Structure (inlined)	0..1	Hyperlink related to the resource. This shall be only included in the HTTP responses and in HTTP PUT requests.
>self	LinkType	1	Self-referring URI. The URI shall be unique within the UE Location Subscribe as it acts as an ID for the subscription.
zoneId	String	1	Identifier of zone (e.g. zone001) to monitor.
upperNumberOfUsersZoneThreshold	UnsignedInt	0..1	Threshold number of users in a zone which if crossed upward shall cause a notification.
lowerNumberOfUsersZoneThreshold	UnsignedInt	0..1	Threshold number of users in a zone which if crossed downward shall cause a notification.
upperNumberOfUsersAPThreshold	UnsignedInt	0..1	Threshold number of users in an access point which if crossed upward shall cause a notification.
lowerNumberOfUsersAPThreshold	UnsignedInt	0..1	Threshold number of users in an access point which if crossed downward shall cause a notification.
operationStatus	OperationStatus	0..N	List of operation status values to generate notifications for (these apply to all access points within a zone). See note 3.
reportingCtrl	ReportingCtrl	0..1	Provides parameters that ctrl the reporting.
expiryDeadline	TimeStamp	0..1	The expiration time of the subscription determined by the Zone Status Service.
NOTE 1:	At least one of callbackReference and websockNotifConfig shall be provided by the service consumer. If both are provided, it is up to location server to select an alternative and return only that alternative in the response, as specified in ETSI GS MEC 009 [4], clause 6.12a.		
NOTE 2:	This allows the client to recover from communication failures during resource creation and therefore avoids duplicate subscription creation in such situations. In case the element is present, the server shall not alter its value, and shall provide it as part of the representation of this resource. In case the element is not present, the server shall not generate it.		

Attribute name	Data type	Cardinality	Description
NOTE 3: As specified in [5], clause 5.2.3.2.			

6.3.8 Type: UserAreaSubscription

This type represents a subscription to the notifications from location server about user location events happened in the area.

The attributes of the UserAreaSubscription shall follow the indications provided in table 6.3.8-1.

Table 6.3.8-1: Definition of type UserAreaSubscription

Attribute name	Data type	Cardinality	Description
subscriptionType	String	1	Shall be set to "UserAreaSubscription".
clientCorrelator	String	0..1	A correlator that the client can use to tag this particular resource representation during a request to create a resource on the server. See note 2.
callbackReference	Uri	0..1	URI exposed by the client on which to receive notifications via HTTP. See note 1.
requestTestNotification	Boolean	0..1	Set to TRUE by the service consumer to request a test notification via HTTP on the callbackReference URI, as specified in ETSI GS MEC 009 [4], clause 6.12a.
websocketNotifConfig	WebsocketNotifConfig	0..1	Provides details to negotiate and signal the use of a WebSocket connection between the location server and the service consumer for notifications. See note 1.
_links	Structure (inlined)	0..1	Hyperlink related to the resource. This shall be only included in the HTTP responses and in HTTP PUT requests.
>self	LinkType	1	Self-referring URI. The URI shall be unique within the UE Location Subscribe as it acts as an ID for the subscription.
areaDefine	ArealInfo	1	The parameters describing the area to subscribe.
addressList	Array(Uri)	1..N	List of the users to be monitored.
trackingAccuracy	Float	1	Number of meters of acceptable error.
locationEventCriteria	Array(LocationEvent Type)	0..N	List of user event values to generate notifications for (these apply to address specified).
reportingCtrl	ReportingCtrl	0..1	Provides parameters that ctrl the reporting.
reportingLocationReq	boolean	0..1	This IE shall be set to true if a location estimate is required for each event report.
expiryDeadline	TimeStamp	0..1	The expiration time of the subscription determined by the UE Area Subscribe Service.
NOTE 1: At least one of callbackReference and websocketNotifConfig shall be provided by the service consumer. If both are provided, it is up to location server to select an alternative and return only that alternative in the response, as specified in ETSI GS MEC 009 [4], clause 6.12a.			
NOTE 2: This allows the client to recover from communication failures during resource creation and therefore avoids duplicate subscription creation in such situations. In case the element is present, the server shall not alter its value, and shall provide it as part of the representation of this resource. In case the element is not present, the server shall not generate it.			

6.3.9 Type: UserDistanceSubscription

This type represents a subscription to the notifications from location server about changes in the geographical relationships between monitored users or between monitored users and reference users.

The attributes of the UserDistanceSubscription shall follow the indications provided in table 6.3.9-1.

Table 6.3.9-1: Definition of type UserDistanceSubscription

Attribute name	Data type	Cardinality	Description
subscriptionType	String	1	Shall be set to "UserDistanceSubscription".

Attribute name	Data type	Cardinality	Description
clientCorrelator	String	0..1	A correlator that the client can use to tag this particular resource representation during a request to create a resource on the server. See note 2.
callbackReference	Uri	0..1	URI exposed by the client on which to receive notifications via HTTP. See note 1.
requestTestNotification	Boolean	0..1	Set to TRUE by the service consumer to request a test notification via HTTP on the callbackReference URI, as specified in ETSI GS MEC 009 [4], clause 6.12a.
websocketNotifConfig	WebsocketNotifConfig	0..1	Provides details to negotiate and signal the use of a WebSocket connection between the location server and the service consumer for notifications. See note 1.
_links	Structure (inlined)	0..1	Hyperlink related to the resource. This shall be only included in the HTTP responses and in HTTP PUT requests.
>self	LinkType	1	Self-referring URI. The URI shall be unique within the User Distance Subscribe as it acts as an ID for the subscription.
referenceAddress	Array (Uri)	0..N	If specified, indicates address of each user that will be used as reference users from which the distances towards monitored users indicated in the Addresses will be monitored (e.g. 'sip' URI, 'tel' URI, 'acr' URI). Reference to a group could be provided here if supported by implementation.
monitoredAddress	Array (Uri)	1..N	Contains addresses of users to monitor (e.g. 'sip' URI, 'tel' URI, 'acr' URI). Reference to a group could be provided here if supported by implementation. If the ReferenceAddress is specified, then the distance between each monitored user and reference user(s) will be monitored. If the ReferenceAddress is not present, then the distance between each pair of the monitored users will be monitored. Note that in that case there shall be at least two addresses specified here.
distance	Float	1	Distance between users that shall be monitored. The unit is meter.
trackingAccuracy	Float	1	Number of meters of acceptable error in tracking distance.
criteria	DistanceCriteria	1	Indicates whether the notification should occur when the geographical relationship between monitored and referenced users' changes. See note 3.
checkImmediate	Bool	1	Check location immediately after establishing notification.
reportingCtrl	ReportingCtrl	0..1	Provides parameters that ctrl the reporting.
expiryDeadline	TimeStamp	0..1	The expiration time of the subscription determined by the UE Distance Subscribe Service.
NOTE 1: At least one of callbackReference and websocketNotifConfig shall be provided by the service consumer. If both are provided, it is up to location server to select an alternative and return only that alternative in the response, as specified in ETSI GS MEC 009 [4], clause 6.12a.			
NOTE 2: This allows the client to recover from communication failures during resource creation and therefore avoids duplicate subscription creation in such situations. In case the element is present, the server shall not alter its value, and shall provide it as part of the representation of this resource. In case the element is not present, the server shall not generate it.			
NOTE 3: As specified in [6], clause 5.2.3.2.			

6.4 Notifications data types

6.4.1 Void

6.4.2 Void

6.4.3 Type: TestNotification

This type represents a test notification from a location server to determine if the Websocket method is to be utilized for the location server to issue notifications for a subscription, as defined in clause 6.12a of ETSI GS MEC 009 [4].

Table 6.4.3-1: Attributes of the TestNotification

Attribute name	Data type	Cardinality	Description
notificationType	String	1	Shall be set to "TestNotification".
_links	Structure (inlined)	1	Hyperlink related to the resource.
>subscription	LinkType	1	URI identifying the subscription for the test notification.

6.4.4 Type: UserLocationEventNotification

This type represents a notification from location server with regards to UE location event. The notification is sent by the location server to inform about the modification of UE location.

The attributes of the UserLocationEventNotification shall follow the indications provided in table 6.4.4-1.

Table 6.4.4-1: Attributes of the UserLocationEventNotification

Attribute name	Data type	Cardinality	Description
notificationType	String	1	Shall be set to "UserLocationEventNotification".
timeStamp	TimeStamp	0..1	Time stamp.
address	Uri	0..1	Address of user (e.g. 'sip' URI, 'tel' URI, 'acr' URI).
userLocationEvent	LocationEventType	1	The specific event triggering this notification, e.g. "ENTERING_AREA_EVENT", "LEAVING_AREA_EVENT".
locationInfo	LocationInfo	0..1	The geographical coordinates where the user is. See note 3.
civicInfo	CivicAddress	0..1	Contextual information of a user location (e.g. aisle, floor, room number, etc.). See notes 1 and 3.
relativeLocationInfo	RelativeLocationInfo	0..1	The relative location in a reference system. See note 3.
zoneld	String	0..1	The identity of the zone. For the events of "ENTERING_AREA_EVENT", it is the zone that the user is currently within. For the event of "LEAVING_AREA_EVENT", it is the zone that the user used to be within. See note 2.
accessPointId	String	0..1	The identity of the access point. For the events of "ENTERING_AREA_EVENT", it indicates the access point that the user is currently within. For the event of "LEAVING_AREA_EVENT", it indicates the access point that the user used to be within. See note 2.
_links	Structure(inlined)	1	Object containing hyperlinks related to the resource.
>subscription	LinkType	1	A link to the related subscription.
NOTE 1: As specified in [17], clause 6.1.6.2.14.			
NOTE 2: As specified in [5], clause 5.2.2.7.			
NOTE 3: At least one of these attributes shall be present only when reportingLocationReq is set to TRUE in the UserLocationEventSubscription.			

6.4.5 Type: UserLocationPeriodicNotification

This type represents a notification from location server with regards to UE location periodic reporting. The notification is sent by the location server periodically for monitoring the UE location.

The attributes of the UserLocationPeriodicNotification shall follow the indications provided in table 6.4.5-1.

Table 6.4.5-1: Attributes of the UserLocationPeriodicNotification

Attribute name	Data type	Cardinality	Description
notificationType	String	1	Shall be set to "UserLocationPeriodicNotification".
timeStamp	TimeStamp	0..1	Time stamp.
address	Uri	0..1	Address of user (e.g. 'sip' URI, 'tel' URI, 'acr' URI).
result	NotificationResult	1	The result of Localization. See note 3.
locationInfo	LocationInfo	0..1	The geographical coordinates where the user is.
civicInfo	CivicAddress	0..1	Contextual information of a user location (e.g. aisle, floor, room number, etc.). See note 1.
relativeLocationInfo	RelativeLocationInfo	0..1	The relative location in a reference system.
zoneId	String	0..1	The identity of the zone that the user is currently within. See note 2.
accessPointId	String	0..1	The identity of the access point that the user is currently within. See note 2.
isFinalNotification	Boolean	0..1	Shall be set to true if it is a final notification.
_links	Structure(inlined)	1	Object containing hyperlinks related to the resource.
>subscription	LinkType	1	A link to the related subscription.
NOTE 1: As specified in [17], clause 6.1.6.2.14.			
NOTE 2: As specified in [5], clause 5.2.2.7.			
NOTE 3: If the result is SUCCESS, at least one of locationInfo, civicInfo and relativeLocationInfo shall be present.			

6.4.6 Type: ZoneLocationEventNotification

This type represents a notification from location server with regards to UE location event happened in a zone. The notification is sent by the location server to inform about the modification of UE location.

The attributes of the ZoneLocationEventNotification shall follow the indications provided in table 6.4.6-1.

Table 6.4.6-1: Attributes of the ZoneLocationEventNotification

Attribute name	Data type	Cardinality	Description
notificationType	String	1	Shall be set to "ZoneLocationEventNotification".
timeStamp	TimeStamp	0..1	Time stamp.
address	Uri	1	Address of user (e.g. 'sip' URI, 'tel' URI, 'acr' URI).
userLocationEvent	LocationEventType	1	The specific event triggering this notification, e.g. "ENTERING_AREA_EVENT", "LEAVING_AREA_EVENT".
zoneId	String	1	The identity of the zone.
_links	Structure(inlined)	1	Object containing hyperlinks related to the resource.
>subscription	LinkType	1	A link to the related subscription.

6.4.7 Type: ZoneStatusNotification

This type represents a notification from location server with regards to status changing event or user number reaching threshold event happened in a zone.

The attributes of the ZoneStatusNotification shall follow the indications provided in table 6.4.7-1.

Table 6.4.7-1: Attributes of the ZoneStatusNotification

Attribute name	Data type	Cardinality	Description
notificationType	String	1	Shall be set to "ZoneStatusNotification".
timeStamp	TimeStamp	0..1	Time stamp.
userNumEvent	Enum(inlined)	0..1	Shall be present when ZoneStatusSubscription includes upperNumberOfUsersZoneThreshold, lowerNumberOfUsersZoneThreshold, upperNumberOfUsersAPThreshold or lowerNumberOfUsersAPThreshold, and the number of users in a zone or an access point crosses the threshold defined in the subscription: 1 = OVER_ZONE_UPPER_THD. 2 = UNDER_ZONE_LOWER_THD. 3 = OVER_AP_UPPER_THD. 4 = UNDER_AP_LOWER_THD.
operationStatus	OperationStatus	0..1	Shall be present when ZoneStatusSubscription includes operationStatus and the operation status value of an access point meets Serviceable or Unserviceable or Unknown defined in the subscription. See note.
zoneld	String	1	The identity of the zone.
accessPointId	String	0..1	Identifier of an access point (e.g. ap01). Shall be included when userNumEvent related with access point or operationStatus is included.
_links	Structure(inlined)	1	Object containing hyperlinks related to the resource.
>subscription	LinkType	1	A link to the related subscription.

NOTE: As specified in [5], clause 5.2.3.2.

6.4.8 Type: UserAreaNotification

This type represents a notification from location server with regards to UE location events happened in an area.

The attributes of the UserAreaNotification shall follow the indications provided in table 6.4.8-1.

Table 6.4.8-1: Attributes of the UserAreaNotification

Attribute name	Data type	Cardinality	Description
notificationType	String	1	Shall be set to "UserAreaNotification".
timeStamp	TimeStamp	0..1	Time stamp.
address	Uri	1	Address of user (e.g. 'sip' URI, 'tel' URI, 'acr' URI).
locationInfo	LocationInfo	0..1	The geographical coordinates where the user is. See note 2.
civicInfo	CivicAddress	0..1	Contextual information of a user location (e.g. aisle, floor, room number, etc.). See notes 1 and 2.
relativeLocationInfo	RelativeLocationInfo	0..1	The relative location in a reference system. See note 2.
userLocationEvent	LocationEventType	1	The specific event triggering this notification, e.g. "ENTERING_AREA_EVENT", "LEAVING_AREA_EVENT".
_links	Structure(inlined)	1	Object containing hyperlinks related to the resource.
>subscription	LinkType	1	A link to the related subscription.

NOTE 1: As specified in [17], clause 6.1.6.2.14.
NOTE 2: At least one of these attributes shall be present only when reportingLocationReq is set to TRUE in the UserAreaSubscription.

6.4.9 Type: UserDistanceNotification

This type represents a notification from location server with regards to changes in the geographical relationships between monitored users or between monitored users and reference users.

The attributes of the UserDistanceNotification shall follow the indications provided in table 6.4.9-1.

Table 6.4.9-1: Attributes of the UserDistanceNotification

Attribute name	Data type	Cardinality	Description
notificationType	String	1	Shall be set to "UserDistanceNotification".
timeStamp	TimeStamp	0..1	Time stamp.
monitoredUsers	UserList	1	Indicates the location information related to monitored users.
distanceEvent	DistanceCriteria	1	Indicates the distance event triggering the notification. See note.
_links	Structure(inlined)	1	Object containing hyperlinks related to the resource.
>subscription	LinkType	1	A link to the related subscription.

NOTE: As specified in [6], clause 5.2.3.2.

6.5 Referenced structured data types

6.5.1 Introduction

This clause defines data structures that are referenced from data structures defined in the previous clauses, but can neither be resource representations nor bound to any pub/sub mechanism.

6.5.2 Type: TimeStamp

This type represents a time stamp.

Table 6.5.2-1: Attributes of type TimeStamp

Attribute name	Data type	Cardinality	Description
seconds	Uint32	1	The seconds part of the time. Time is defined as: Unix-time since January 1, 1970, 00:00:00 UTC.
nanoSeconds	Uint32	1	The nanoseconds part of the time. Time is defined as: Unix-time since January 1, 1970, 00:00:00 UTC.

6.5.3 Type: LocationInfo

This data type extends the LocationInfo data type defined in [5], clause 5.2.2.5, and its elements shall follow the notations provided in table 6.5.3-1. The World Geodetic System 1984 (WGS 84) [13] shall be used as the base coordinate reference system for the coding of location information. The listed attributes are aligned to those defined for the Universal Geographical Area Description (GAD) specified in ETSI TS 123 032 [14]. However, attributes are not limited to the reporting of location information obtained from a 3GPP based access network. For instance, location information from WLAN may also include uncertainty information [15]. In addition, the source of the location information is not limited to GNSS reports from the end user device and may be provided an entity associated with the underlying access network, or even by a specific MEC application.

Table 6.5.3-1: Definition of type LocationInfo

Attribute name	Data type	Cardinality	Description
latitude	Float	1..N	Location latitude, expressed in the range -90° to +90°. Cardinality greater than one only if "shape" equals 7.
longitude	Float	1..N	Location longitude, expressed in the range -180° to +180°. Cardinality greater than one only if "shape" equals 7.
altitude	Float	0..1	Location altitude relative to the WGS84 ellipsoid surface.
accuracy	UnsignedInt	0..1	Horizontal accuracy/(semi-major) uncertainty of location provided in meters, as defined in [14]. Present only if "shape" equals 4, 5 or 6.

Attribute name	Data type	Cardinality	Description
accuracySemiMinor	UnsignedInt	0..1	Horizontal accuracy/(semi-major) uncertainty of location provided in meters, as defined in [14]. Present only if "shape" equals 4, 5 or 6.
accuracyAltitude	UnsignedInt	0..1	Altitude accuracy/uncertainty of location provided in meters, as defined in [14]. Present only if "shape" equals 3 or 4.
orientationMajorAxis	UnsignedInt	0..1	Angle of orientation of the major axis, expressed in the range 0° to 180°, as defined in [14]. Present only if "shape" equals 4 or 6.
confidence	UnsignedInt	0..1	Confidence by which the position of a target entity is known to be within the shape description, expressed as a percentage and defined in [14]. Present only if "shape" equals 1, 4 or 6.
innerRadius	UnsignedInt	0..1	Present only if "shape" equals 6.
uncertaintyRadius	UnsignedInt	0..1	Present only if "shape" equals 6.
offsetAngle	UnsignedInt	0..1	Present only if "shape" equals 6.
includedAngle	UnsignedInt	0..1	Present only if "shape" equals 6.
shape	Enum (inlined)	1	Shape information, as detailed in [14], associated with the reported location coordinate: 1 = ELLIPSOID_ARC. 2 = ELLIPSOID_POINT. 3 = ELLIPSOID_POINT_ALTITUDE. 4 = ELLIPSOID_POINT_ALTITUDE_UNCERT_ELLIPSOID. 5 = ELLIPSOID_POINT_UNCERT_CIRCLE. 6 = ELLIPSOID_POINT_UNCERT_ELLIPSE. 7 = POLYGON.
velocity	Structure (inlined)	0..1	Structure with attributes relating to the target entity's velocity, as defined in [14].
>velocityType	Enum (inlined)	1	Velocity information, as detailed in [14], associated with the reported location coordinate: 1 = HORIZONTAL. 2 = HORIZONTAL_VERTICAL. 3 = HORIZONTAL_UNCERT. 4 = HORIZONTAL_VERTICAL_UNCERT.
>bearing	UnsignedInt	1	Bearing, expressed in the range 0° to 360°, as defined in [14].
>horizontalSpeed	UnsignedInt	1	Horizontal speed, expressed in km/h and defined in [14].
>uncertainty	UnsignedInt	0..1	Horizontal uncertainty, as defined in [14]. Present only if "velocityType" equals 3 or 4.
>verticalSpeed	Int	0..1	Vertical speed, expressed in km/h and defined in [14]. Present only if "velocityType" equals 2 or 4.
>verticalUncertainty	UnsignedInt	0..1	Vertical uncertainty, as defined in [14]. Present only if "velocityType" equals 4.

6.5.4 Type: WebsocketNotifConfig

This type represents configuration for the delivery of subscription notifications over Websockets per the pattern defined in defined in clause 6.12a of ETSI GS MEC 009 [4].

Table 6.5.4-1: Attributes of the WebsocketNotifConfig

Attribute name	Data type	Cardinality	Description
websocketUri	Uri	0..1	Set by location server to indicate to the service consumer the Websocket URI to be used for delivering notifications.
requestWebsocketUri	Boolean	0..1	Set to true by the service consumer to indicate that Websocket delivery is requested.

6.5.5 Type: UserEventPara

This type represents some requirements about reporting the event of user location.

Table 6.5.5-1: Attributes of the UserEventPara

Attribute name	Data type	Cardinality	Description
reportingLocationReq	boolean	0..1	This IE shall be set to true if a location estimate is required for each event report.
accessPointList	Array(String)	0..N	One or more access points forming a monitoring area that could be any shape. See note 1.
zoneId	String	0..1	Identifier of zone (e.g. zone001) to monitor. See note 1.
occurrenceInfo	OccurrenceInfo	0..1	One time only report indication. See note 2.
NOTE 1: Only one of accessPointList and zoneId may be present.			
NOTE 2: As specified in [17], clause 6.1.6.3.16.			

6.5.6 Type: ReportingCtrl

This type represents the parameters that control the report times and frequency.

Table 6.5.6-1: Attributes of type ReportingCtrl

Attribute name	Data type	Cardinality	Description
minimumInterval	UnsignedInt	0..1	Minimum interval between reports in case frequently reporting. Unit is second.
maximumFrequency	UnsignedInt	0..1	Maximum frequency (in seconds) of notifications per subscription.
maximumCount	UnsignedInt	0..1	Maximum number of notifications. For no maximum, either do not include this element or specify a value of zero. Default value is 0.

6.5.7 Type: ArealInfo

This type represents the parameters that describe an area.

Table 6.5.7-1: Attributes of type ArealInfo

Attribute name	Data type	Cardinality	Description
shape	Enum(inlined)	1	The shape of the area monitored: 1 = CIRCLE. 2 = POLYGON.
points	Array(Point)	1..N	Shall include one point if the shape is CIRCLE. Shall include 3-15 points if the shape is POLYGON.
radius	UnsignedInt	0..1	Shall be present if the shape is CIRCLE.

6.5.8 Type: Point

This type represents the geographical location of a point.

Table 6.5.8-1: Attributes of type Point

Attribute name	Data type	Cardinality	Description
latitude	Float	1	Location latitude, expressed in the range -90° to +90°.
longitude	Float	1	Location longitude, expressed in the range -180° to +180°.

6.6 Mapping of Identifiers

6.6.1 accessPointId

With respect to identifiers assigned to access points in a cellular deployment, the accessPointId shall equal to an E-CGI concatenated with a Cell Portion ID as defined in ETSI TS 129 171 [8], i.e.:

- accessPointId = <E-CGI><Cell Portion ID>

where the Cell Portion ID may not exist, if the cell does not contain portions.

6.6.2 userId

The Location Service allows using the UE's IP address as a user identifier. The OMA API [5] does not explicitly list IP address as a user identifier option (i.e. as part of the "address" Element of UserInfo). Therefore to satisfy the requirements of the Location Service, if a user identifier of type anyURI is in the form of an IP address, the "acr" scheme shall be used, followed by the UE's IP address, i.e.:

- address = acr:<UE's IP address>

where the format of the UE's IP address shall be an IP literal encapsulated within square brackets or an Ipv4 address in dotted-decimal form.

NOTE: The "acr" scheme can also be used to convey user identifiers other than an IP address, e.g. an alias.

6.7 Referenced simple data types and enumerations

6.7.1 Introduction

This clause defines simple data types and enumerations that can be referenced from data structures defined in the previous clauses.

6.7.2 Type: LinkType

This type represents a type of link and may be referenced from data structures.

Table 6.7.2-1: Definition of type LinkType

Attribute name	Data type	Cardinality	Description
href	Uri	1	URI referring to a resource.

6.7.3 Enumeration: LocationEventType

This type represents specified event types for UE location report.

Table 6.7.3-1: Enumeration LocationEventType

Type name	Description
"ENTERING_AREA_EVENT"	Entering area reporting event.
"LEAVING_AREA_EVENT"	Leaving area reporting event.

6.7.4 Enumeration: NotificationResult

This enumeration represents the result of a localization associated with a notification.

Table 6.7.4-1: Enumeration NotificationResult

Type name	Description
SUCCESS	Localization is successful.
ABNORMAL	Localization is failed due to timeout or other unspecific errors.

6.7.5 Enumeration: ConnectionType

This enumeration represents the connection type of an access point.

Table 6.7.5-1: Enumeration ConnectionType

Type name	Description
LTE	Access point provides a LTE-Femto connection.
Wi-Fi®	Access point provides a Wi-Fi® connection.
WiMAX®	Access point provides a WiMAX® connection.
5G NR	Access point provides a 5G NR connection.
UNKNOWN	Access point connection type is unknown.

7 API definition

7.1 Introduction

This clause defines the resources and operations of the Location Service API.

7.2 Global definitions and resource structure

All resource URLs of this API shall have the following root:

{apiRoot}/location/{apiVersion}/

"apiRoot" is discovered using the service registry, it includes the scheme ("http" or "https"), host and optional port, and an optional prefix string. The "apiVersion" shall be set to "v2" for the current version of the present document. It includes the scheme ("http" or "https"), host and optional port, and an optional prefix string.

This API shall support HTTP over TLS (also known as HTTPS) using TLS version 1.2 as defined by IETF RFC 5246 [10]. TLS 1.3 (including the new specific requirements for TLS 1.2 implementations) defined by IETF RFC 8446 [16] should be supported. HTTP without TLS shall not be used. Versions of TLS earlier than 1.2 shall neither be supported nor used.

The content format of JSON shall be supported and signalled by setting the content type header to "application/json".

This API supports additional application-related error information to be provided in the HTTP response when an error occurs. See clause 6.15 of ETSI GS MEC 009 [4] for more information.

This API shall require the use of the OAuth 2.0 client credentials grant type according to IETF RFC 6749 [11] with bearer tokens according to IETF RFC 6750 [12]. See ETSI GS MEC 009 [4], clause 6.16 for more information. How the token endpoint and client credentials are provisioned into the MEC applications is out of scope of the present document.

Figure 7.2-1 illustrates the resource URI structure of this API.

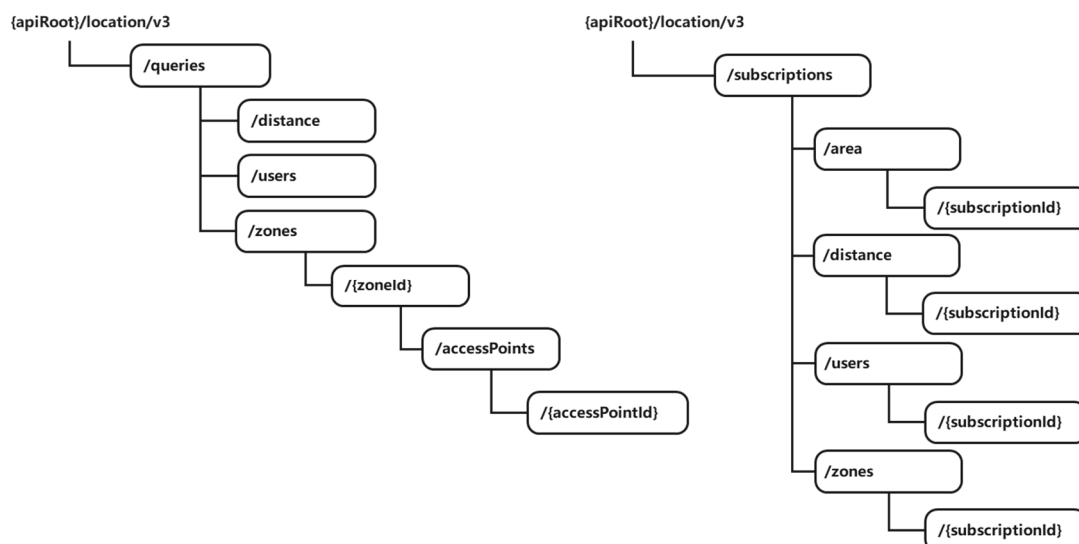


Figure 7.2-1: Resource URI structure of the Location API

Table 7.2-1 provides an overview of the resources defined by the present specification for the MEC location API, and the applicable HTTP methods.

Table 7.2-1: Resources and methods overview

Resource name	Resource URI	HTTP method	Meaning
users	/queries/users	GET	UE Location Lookup of a specific UE or group of UEs.
zones	/queries/zones	GET	Zones information Lookup.
individual zone	/queries/zones/{zoneId}	GET	Zones information Lookup.
accessPoints	/queries/zones/{zoneId}/accessPoints	GET	Radio Node Location Lookup.
individual accessPoint	/queries/zones/{zoneId}/accessPoints/{accessPointId}	GET	Radio Node Location Lookup.
distance	/queries/distance/	GET	UE Distance Lookup of a specific UE.
user_subscriptions	/subscriptions/users/	GET	Retrieves information about the subscriptions for the requestor.
		POST	Create subscription to UE location.
Individual user_subscription	subscriptions/users/{subscriptionId}	GET	Retrieve subscription information.
		PUT	Updates a subscription information.
		DELETE	Delete a subscription.
zone_subscriptions	/subscriptions/zones	GET	Retrieves all active subscriptions to zone notifications.
		POST	Creates a subscription to zone notifications.
individual zone_subscription	/subscriptions/zones/{subscriptionId}	GET	Retrieve zone subscription information.
		PUT	Updates a zone subscription information.
		DELETE	Cancel a zone subscription.
distance_subscriptions	/subscriptions/distance	GET	Retrieves all active subscriptions to distance change notifications.
		POST	Creates a subscription for distance change notification.
individual distance_subscription	/subscriptions/distance/{subscriptionId}	GET	Retrieve user distance subscription information.
		PUT	Updates a user distance subscription information.
		DELETE	Cancel a subscription.

Resource name	Resource URI	HTTP method	Meaning
area_subscriptions	/subscriptions/area	GET	Retrieves information about the subscriptions for this requestor.
		POST	Creates subscription to area notifications.
individual area_subscription	/subscriptions/area/{subscriptionId}	GET	Retrieve subscription information.
		PUT	Updates a subscription information.
		DELETE	Cancel a subscription.

7.3 Void

7.4 Resource: users

7.4.1 Description

This resource is queried to retrieve location information about a specific UE or a group of UEs.

7.4.2 Resource definition

Resource URI: {apiRoot}/location/v3/queries/users

This resource shall support the resource URI variables defined in table 7.4.2-1.

Table 7.4.2-1: Resource URI variables for resource "users"

Name	Definition
apiRoot	See clause 7.2.

7.4.3 Resource methods

7.4.3.1 GET

The GET method is used to query location information about a specific UE or a group of UEs.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in tables 7.4.3.1-1 and 7.4.3.1-2.

Table 7.4.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
zoneld	String	0..N	Identifier of zone (e.g. zone001).
accessPointId	String	0..N	Identifier of access point (e.g. 00101000000000000000000000000001).
address	AnyUri	0..N	Address of users (e.g. 'sip' URI, 'tel' URI, 'acr' URI). See note.
NOTE:	In order to perform the UE Location Lookup for a specific UE or a group of specific UEs, the filtering parameter "address" shall be used.		

Table 7.4.3.1-2: Data structures supported by the GET request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	Data type	Cardinality	Response Codes	Remarks
	UserList	1	200 OK	Upon success, a response body containing the list of user location information is returned.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.4.3.2 PUT

Not applicable.

7.4.3.3 PATCH

Not applicable.

7.4.3.4 POST

Not applicable.

7.4.3.5 DELETE

Not applicable.

7.5 Resource: user_subscriptions

7.5.1 Description

This resource contains various resources related to subscriptions for notifications related to location.

7.5.2 Resource definition

Resource URI: **{apiRoot}/location/v3/subscriptions/users/**

This resource shall support the resource URI variables defined in table 7.5.2-1.

Table 7.5.2-1: Resource URI variables for resource "user_subscriptions"

Name	Definition
apiRoot	See clause 7.2.

7.5.3 Resource methods

7.5.3.1 GET

The GET method is used to request information about the subscriptions for this requestor. Upon success, the response contains entity body with the list of links to the subscriptions that are present for the requestor.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in tables 7.5.3.1-1 and 7.5.3.1-2.

Table 7.5.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
subscription_type	String	0..1	Query parameter to filter on a specific subscription type. Permitted values: <ul style="list-style-type: none"> event. periodic.
address	Uri	0..1	Address of user (e.g. 'sip' URI, 'tel' URI, 'acr' URI).

Table 7.5.3.1-2: Data structures supported by the GET request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	Data type	Cardinality	Response Codes	Remarks
	NotificationSubscriptionList	1	200 OK	Upon success, a response body containing the list of links to requestor's subscriptions is returned.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.5.3.2 PUT

Not applicable.

7.5.3.3 PATCH

Not applicable.

7.5.3.4 POST

The POST method is used to create a new subscription to UE location notifications. Upon success, the response contains entity body describing the created subscription.

This method shall support the request and response data structures, and response codes, as specified in table 7.5.3.4-1.

Table 7.5.3.4-1: Data structures supported by the POST request/response on this resource

Request body	Data type	Cardinality	Remarks	
	{NotificationSubscription}	1	The entity body in the request contains data type of the specific UE location event subscription that is to be created, where the data type options are listed below and defined in clauses 6.3.4 and 6.3.5: <ul style="list-style-type: none"> UserLocationEventSubscription. UserLocationPeriodicSubscription. 	
Response body	Data type	Cardinality	Response Codes	Remarks
	{NotificationSubscription}	1	201 Created	Indicates successful resource creation, where the resource URI shall be returned in the HTTP Location header field. In the returned NotificationSubscription structure, the created subscription is described using the appropriate data type from the list below and as defined in clauses 6.3.4 and 6.3.5: <ul style="list-style-type: none"> UserLocationEventSubscription. UserLocationPeriodicSubscription.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	415 Unsupported Media Type	It is used to indicate that the server or the client does not support the content type of the entity body. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

	ProblemDetails	0..1	422 Unprocessable Entity	It is used to indicate that the server understands the content type of the request entity and that the syntax of the request entity is correct but that the server is unable to process the contained instructions. This error condition can occur if an JSON request body is syntactically correct but semantically incorrect, for example if the target area for the request is considered too large. This error condition can also occur if the capabilities required by the request are not supported. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.5.3.5 DELETE

Not applicable.

7.6 Resource: individual user_subscription

7.6.1 Description

This resource represents a subscription that the client has created to receive location event notifications.

7.6.2 Resource definition

Resource URI: **{apiRoot}/location/v3/subscriptions/users/{subscriptionId}**

This resource shall support the resource URI variables defined in table 7.6.2-1.

Table 7.6.2-1: Resource URI variables for resource "individual user_subscription"

Name	Definition
apiRoot	See clause 7.2.
subscriptionId	Refers to created subscription, where the Location API allocates a unique resource name for this subscription. The resource name can be also used to identify the resource.

7.6.3 Resource methods

7.6.3.1 GET

The GET method is used to retrieve information about this subscription. Upon success, the response contains entity body with the data type describing the subscription.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in tables 7.6.3.1-1 and 7.6.3.1-2.

Table 7.6.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.6.3.1-2: Data structures supported by the GET request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	Data type	Cardinality	Response Codes	Remarks
	{NotificationSubscription}	1	200 OK	Upon success, a response body containing data type describing the specific Location event subscription is returned. The allowed data types for subscriptions are defined in clauses 6.3.4 and 6.3.5: <ul style="list-style-type: none"> UserLocationEventSubscription. UserLocationPeriodicSubscription.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.6.3.2 PUT

The PUT method is used to update the existing subscription. PUT method in this case has "replace" semantics. Upon successful operation, the target resource is updated with new Data Type received within the message body of the PUT request.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in tables 7.6.3.2-1 and 7.6.3.2-2.

Table 7.6.3.2-1: URI query parameters supported by the PUT method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.6.3.2-2: Data structures supported by the PUT request/response on this resource

Request body	Data type	Cardinality	Remarks	
	{NotificationSubscription}	1	New NotificationSubscription is included as entity body of the request. The allowed data types for subscriptions are defined in clauses 6.3.4 and 6.3.5: <ul style="list-style-type: none"> UserLocationEventSubscription. UserLocationPeriodicSubscription. 	
Response body	Data type	Cardinality	Response Codes	Remarks
	{NotificationSubscription}	1	200 OK	Upon success, a response body containing data type describing the updated subscription is returned. The allowed data types for subscriptions are defined in clauses 6.3.4 and 6.3.5: <ul style="list-style-type: none"> UserLocationEventSubscription. UserLocationPeriodicSubscription.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	412 Precondition Failed	It is used when a condition has failed during conditional requests, e.g. when using Etags to avoid write conflicts when using PUT. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

	ProblemDetails	0..1	422 Unprocessable Entity	It is used to indicate that the server understands the content type of the request entity and that the syntax of the request entity is correct but that the server is unable to process the contained instructions. This error condition can occur if a JSON request body is syntactically correct but semantically incorrect, for example if the target area for the request is considered too large. This error condition can also occur if the capabilities required by the request are not supported. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.6.3.3 PATCH

Not applicable.

7.6.3.4 POST

Not applicable.

7.6.3.5 DELETE

The DELETE method is used to cancel the existing subscription. Cancellation can be made by deleting the resource that represents existing subscription.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in tables 7.6.3.5-1 and 7.6.3.5-2.

Table 7.6.3.5-1: URI query parameters supported by the DELETE method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.6.3.5-2: Data structures supported by the DELETE request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	Data type	Cardinality	Response Codes	Remarks
	n/a		204 No Content	Upon success, a response 204 No Content without any response body is returned.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.7 Resource: zones

7.7.1 Description

This resource is queried to retrieve the information about a specific zone or a list of zones.

7.7.2 Resource definition

Resource URI: {apiRoot}/location/v3/queries/zones

This resource shall support the resource URI variables defined in table 7.7.2-1.

Table 7.7.2-1: Resource URI variables for resource "zones"

Name	Definition
apiRoot	See clause 7.2.

7.7.3 Resource methods

7.7.3.1 GET

The GET method is used to query the information about one or more specific zones or a list of zones.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in tables 7.7.3.1-1 and 7.7.3.1-2.

Table 7.7.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
zoneId	String	0..N	Identifier of zone (e.g. zone001).

Table 7.7.3.1-2: Data structures supported by the GET request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	Data type	Cardinality	Response Codes	Remarks
	ZoneList	1	200 OK	Upon success, a response body containing the list of zone information is returned.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.7.3.2 PUT

Not applicable.

7.7.3.3 PATCH

Not applicable.

7.7.3.4 POST

Not applicable.

7.7.3.5 DELETE

Not applicable.

7.8 Resource: individual zone

7.8.1 Description

This resource is queried to retrieve the information about a specific zone.

7.8.2 Resource definition

Resource URI: {apiRoot}/location/v3/queries/zones/{zoneId}

This resource shall support the resource URI variables defined in table 7.8.2-1.

Table 7.8.2-1: Resource URI variables for resource "individual zone"

Name	Definition
apiRoot	See clause 7.2.

7.8.3 Resource methods

7.8.3.1 GET

The GET method is used to query the information about a specific zone.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in tables 7.8.3.1-1 and 7.8.3.1-2.

Table 7.8.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.8.3.1-2: Data structures supported by the GET request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	Data type	Cardinality	Response Codes	Remarks
	ZoneInfo	1	200 OK	Upon success, a response body containing the zone information is returned.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
ProblemDetails	1	403 Forbidden		The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.

	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.8.3.2 PUT

Not applicable.

7.8.3.3 PATCH

Not applicable.

7.8.3.4 POST

Not applicable.

7.8.3.5 DELETE

Not applicable.

7.9 Resource: accessPoints

7.9.1 Description

This resource is queried to retrieve the information about one or more specific access points or a list of access points under a zone.

7.9.2 Resource definition

Resource URI: **{apiRoot}/location/v3/queries/zones/{zoneId}/accessPoints**

This resource shall support the resource URI variables defined in table 7.9.2-1.

Table 7.9.2-1: Resource URI variables for resource "accessPoints"

Name	Definition
apiRoot	See clause 7.2.

7.9.3 Resource methods

7.9.3.1 GET

The GET method is used to query the information about a specific access point or a list of access points under a zone.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in tables 7.9.3.1-1 and 7.9.3.1-2.

Table 7.9.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
accessPointId	String	0..N	Identifier of access point (e.g. 00101000000000000000000000000001).

Table 7.9.3.1-2: Data structures supported by the GET request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	Data type	Cardinality	Response Codes	Remarks
	AccessPointList	1	200 OK	Upon success, a response body containing the list of access point information is returned.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.9.3.2 PUT

Not applicable.

7.9.3.3 PATCH

Not applicable.

7.9.3.4 POST

Not applicable.

7.9.3.5 DELETE

Not applicable.

7.10 Resource: individual accessPoint

7.10.1 Description

This resource is queried to retrieve the information about a specific access point under a zone.

7.10.2 Resource definition

Resource URI: **{apiRoot}/location/v3/queries/zones/{zoneId}/accessPoints/{accessPointId}**

This resource shall support the resource URI variables defined in table 7.10.2-1.

Table 7.10.2-1: Resource URI variables for resource "individual accessPoint"

Name	Definition
apiRoot	See clause 7.2.

7.10.3 Resource methods

7.10.3.1 GET

The GET method is used to query the information about a specific access point under a zone.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in tables 7.10.3.1-1 and 7.10.3.1-2.

Table 7.10.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.10.3.1-2: Data structures supported by the GET request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	Data type	Cardinality	Response Codes	Remarks
	AccessPointInfo	1	200 OK	Upon success, a response body containing the access point information is returned.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.10.3.2 PUT

Not applicable.

7.10.3.3 PATCH

Not applicable.

7.10.3.4 POST

Not applicable.

7.10.3.5 DELETE

Not applicable.

7.11 Resource: zone_subscriptions

7.11.1 Description

This resource contains various resources related to subscriptions for notifications related to location.

7.11.2 Resource definition

Resource URI: {apiRoot}/location/v3/subscriptions/zones

This resource shall support the resource URI variables defined in table 7.11.2-1.

Table 7.11.2-1: Resource URI variables for resource "zone_subscriptions"

Name	Definition
apiRoot	See clause 7.2.

7.11.3 Resource methods

7.11.3.1 GET

The GET method is used to request information about the subscriptions for this requestor. Upon success, the response contains entity body with the list of links to the subscriptions that are present for the requestor.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in tables 7.11.3.1-1 and 7.11.3.1-2.

Table 7.11.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
subscription_type	String	0..1	Query parameter to filter on a specific subscription type. Permitted values: <ul style="list-style-type: none"> event. status.
zoneId	String	0..1	The identity of the zone.

Table 7.11.3.1-2: Data structures supported by the GET request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	Data type	Cardinality	Response Codes	Remarks
	NotificationSubscriptionList	1	200 OK	Upon success, a response body containing the list of links to requestor's subscriptions is returned.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.	

	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.11.3.2 PUT

Not applicable.

7.11.3.3 PATCH

Not applicable.

7.11.3.4 POST

The POST method is used to create a new subscription to zone notifications. Upon success, the response contains entity body describing the created subscription.

This method shall support the request and response data structures, and response codes, as specified in table 7.11.3.4-1.

Table 7.11.3.4-1: Data structures supported by the POST request/response on this resource

	Data type	Cardinality	Remarks	
Request body	{NotificationSubscription}	1	The entity body in the request contains data type of the specific UE location event subscription that is to be created, where the data type options are listed below and defined in clauses 6.3.6 and 6.3.7: <ul style="list-style-type: none"> • ZoneLocationEventSubscription. • ZoneStatusSubscription. 	
	Data type	Cardinality	Response Codes	Remarks
Response body	{NotificationSubscription}	1	201 Created	Indicates successful resource creation, where the resource URI shall be returned in the HTTP Location header field. In the returned NotificationSubscription structure, the created subscription is described using the appropriate data type from the list below and as defined in clauses 6.3.6 and 6.3.7: <ul style="list-style-type: none"> • ZoneLocationEventSubscription. • ZoneStatusSubscription.

ProblemDetails	0..1	400 Bad Request	<p>It is used to indicate that incorrect parameters were passed to the request.</p> <p>In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.</p>
ProblemDetails	0..1	401 Unauthorized	<p>It is used when the client did not submit credentials.</p> <p>In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.</p>
ProblemDetails	1	403 Forbidden	<p>The operation is not allowed given the current status of the resource.</p> <p>More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.</p>
ProblemDetails	0..1	404 Not Found	<p>It is used when a client provided a URI that cannot be mapped to a valid resource URI.</p> <p>In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.</p>
ProblemDetails	0..1	406 Not Acceptable	<p>It is used to indicate that the server cannot provide the any of the content formats supported by the client.</p> <p>In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.</p>
ProblemDetails	0..1	415 Unsupported Media Type	<p>It is used to indicate that the server or the client does not support the content type of the entity body.</p> <p>In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.</p>
ProblemDetails	0..1	422 Unprocessable Entity	<p>It is used to indicate that the server understands the content type of the request entity and that the syntax of the request entity is correct but that the server is unable to process the contained instructions. This error condition can occur if an JSON request body is syntactically correct but semantically incorrect, for example if the target area for the request is considered too large. This error condition can also occur if the capabilities required by the request are not supported.</p> <p>In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.</p>
ProblemDetails	0..1	429 Too Many Requests	<p>It is used when a rate limiter has triggered.</p> <p>In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.</p>

7.11.3.5 DELETE

Not applicable.

7.12 Resource: individual zone_subscription

7.12.1 Description

This resource represents a subscription that the client has created to receive location event notifications.

7.12.2 Resource definition

Resource URI: {apiRoot}/location/v3/subscriptions/zones/{subscriptionId}

This resource shall support the resource URI variables defined in table 7.12.2-1.

Table 7.12.2-1: Resource URI variables for resource "individual zone_subscription"

Name	Definition
apiRoot	See clause 7.2.
subscriptionId	Refers to created subscription, where the Location API allocates a unique resource name for this subscription. The resource name can be also used to identify the resource.

7.12.3 Resource methods

7.12.3.1 GET

The GET method is used to retrieve information about this subscription. Upon success, the response contains entity body with the data type describing the subscription.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in tables 7.12.3.1-1 and 7.12.3.1-2.

Table 7.12.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.12.3.1-2: Data structures supported by the GET request/response on this resource

Request body	Data type	Cardinality	Remarks	
n/a				
Response body	Data type	Cardinality	Response Codes	Remarks
	{NotificationSubscription}	1	200 OK	Upon success, a response body containing data type describing the specific zone subscription is returned. The allowed data types for subscriptions are defined in clauses 6.3.6 and 6.3.7: <ul style="list-style-type: none"> ZoneLocationEventSubscription. ZoneStatusSubscription.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.12.3.2 PUT

The PUT method is used to update the existing subscription. PUT method in this case has "replace" semantics. Upon successful operation, the target resource is updated with new Data Type received within the message body of the PUT request.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in tables 7.12.3.2-1 and 7.12.3.2-2.

Table 7.12.3.2-1: URI query parameters supported by the PUT method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.12.3.2-2: Data structures supported by the PUT request/response on this resource

Request body	Data type	Cardinality	Remarks	
	{NotificationSubscription}	1	New NotificationSubscription is included as entity body of the request. The allowed data types for subscriptions are defined in clauses 6.3.6 and 6.3.7: <ul style="list-style-type: none"> • ZoneLocationEventSubscription. • ZoneStatusSubscription. 	
Response body	Data type	Cardinality	Response Codes	Remarks
	{NotificationSubscription}	1	200 OK	Upon success, a response body containing data type describing the updated subscription is returned. The allowed data types for subscriptions are defined in clauses 6.3.6 and 6.3.7: <ul style="list-style-type: none"> • ZoneLocationEventSubscription. • ZoneStatusSubscription.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

	ProblemDetails	0..1	401 Unauthorized	<p>It is used when the client did not submit credentials.</p> <p>In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.</p>
	ProblemDetails	1	403 Forbidden	<p>The operation is not allowed given the current status of the resource.</p> <p>More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.</p>
	ProblemDetails	0..1	404 Not Found	<p>It is used when a client provided a URI that cannot be mapped to a valid resource URI.</p> <p>In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.</p>
	ProblemDetails	0..1	406 Not Acceptable	<p>It is used to indicate that the server cannot provide the any of the content formats supported by the client.</p> <p>In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.</p>
	ProblemDetails	0..1	412 Precondition Failed	<p>It is used when a condition has failed during conditional requests, e.g. when using Etags to avoid write conflicts when using PUT.</p> <p>In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.</p>
	ProblemDetails	0..1	422 Unprocessable Entity	<p>It is used to indicate that the server understands the content type of the request entity and that the syntax of the request entity is correct but that the server is unable to process the contained instructions. This error condition can occur if an JSON request body is syntactically correct but semantically incorrect, for example if the target area for the request is considered too large. This error condition can also occur if the capabilities required by the request are not supported.</p> <p>In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.</p>
	ProblemDetails	0..1	429 Too Many Requests	<p>It is used when a rate limiter has triggered.</p> <p>In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.</p>

7.12.3.3 PATCH

Not applicable.

7.12.3.4 POST

Not applicable.

7.12.3.5 DELETE

The DELETE method is used to cancel the existing subscription. Cancellation can be made by deleting the resource that represents existing subscription.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in tables 7.12.3.5-1 and 7.12.3.5-2.

Table 7.12.3.5-1: URI query parameters supported by the DELETE method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.12.3.5-2: Data structures supported by the DELETE request/response on this resource

Request body	Data type	Cardinality	Remarks	
n/a				
Response body	Data type	Cardinality	Response Codes	Remarks
	n/a		204 No Content	Upon success, a response 204 No Content without any response body is returned.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.13 Resource: distance

7.13.1 Description

This resource is queried to retrieve information about distance from a user to a location or between two users.

7.13.2 Resource definition

Resource URI: {apiRoot}/location/v3/queries/distance

This resource shall support the resource URI variables defined in table 7.13.2-1.

Table 7.13.2-1: Resource URI variables for resource "distance"

Name	Definition
apiRoot	See clause 7.2.

7.13.3 Resource methods

7.13.3.1 GET

The GET method is used to query information about distance from a user to a location or between two users.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in tables 7.13.3.1-1 and 7.13.3.1-2.

Table 7.13.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
address	Array (Uri)	1..2	One or two addresses of user to check. The second "address" parameter shall not be included when the distance between a user and a location is requested. The second "address" parameter shall be included when a location is not provided.
location	Point	0..1	Shall not be included when the distance between two terminals is requested.

Table 7.13.3.1-2: Data structures supported by the GET request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	Data type	Cardinality	Response Codes	Remarks
	TerminalDistance	1	200 OK	Upon success, a response body containing the list of distance information is returned.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.13.3.2 PUT

Not applicable.

7.13.3.3 PATCH

Not applicable.

7.13.3.4 POST

Not applicable.

7.13.3.5 DELETE

Not applicable.

7.14 Resource: distance_subscriptions

7.14.1 Description

This resource contains various resources related to subscriptions for notifications related to distance changing.

7.14.2 Resource definition

Resource URI: {apiRoot}/location/v3/subscriptions/distance

This resource shall support the resource URI variables defined in table 7.14.2-1.

Table 7.14.2-1: Resource URI variables for resource "distance_subscriptions"

Name	Definition
apiRoot	See clause 7.2.

7.14.3 Resource methods

7.14.3.1 GET

The GET method is used to request information about the subscriptions for this requestor. Upon success, the response contains entity body with the list of links to the subscriptions that are present for the requestor.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in tables 7.14.3.1-1 and 7.14.3.1-2.

Table 7.14.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.14.3.1-2: Data structures supported by the GET request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	Data type	Cardinality	Response Codes	Remarks
	NotificationSubscriptionList	1	200 OK	Upon success, a response body containing the list of links to requestor's subscriptions is returned.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.14.3.2 PUT

Not applicable.

7.14.3.3 PATCH

Not applicable.

7.14.3.4 POST

The POST method is used to create a new subscription to user distance notifications. Upon success, the response contains entity body describing the created subscription.

This method shall support the request and response data structures, and response codes, as specified in table 7.14.3.4-1.

Table 7.14.3.4-1: Data structures supported by the POST request/response on this resource

Request body	Data type	Cardinality	Remarks	
	{NotificationSubscription}	1	The entity body in the request contains data type of the user distance subscription that is to be created, where the data type options are listed below and defined in clause 6.3.9: <ul style="list-style-type: none"> UserDistanceSubscription. 	
Response body	Data type	Cardinality	Response Codes	Remarks
	{NotificationSubscription}	1	201 Created	Indicates successful resource creation, where the resource URI shall be returned in the HTTP Location header field. In the returned NotificationSubscription structure, the created subscription is described using the appropriate data type from the list below and as defined in clause 6.3.9: <ul style="list-style-type: none"> UserDistanceSubscription.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	415 Unsupported Media Type	It is used to indicate that the server or the client does not support the content type of the entity body. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

	ProblemDetails	0..1	422 Unprocessable Entity	It is used to indicate that the server understands the content type of the request entity and that the syntax of the request entity is correct but that the server is unable to process the contained instructions. This error condition can occur if an JSON request body is syntactically correct but semantically incorrect, for example if the target area for the request is considered too large. This error condition can also occur if the capabilities required by the request are not supported. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.14.3.5 DELETE

Not applicable.

7.15 Resource: individual distance_subscription

7.15.1 Description

This resource represents a subscription that the client has created to receive distance event notifications.

7.15.2 Resource definition

Resource URI: {apiRoot}/location/v3/subscriptions/distance/{subscriptionId}

This resource shall support the resource URI variables defined in table 7.15.2-1.

Table 7.15.2-1: Resource URI variables for resource "individual distance_subscription"

Name	Definition
apiRoot	See clause 7.2.
subscriptionId	Refers to created subscription, where the Location API allocates a unique resource name for this subscription. The resource name can be also used to identify the resource.

7.15.3 Resource methods

7.15.3.1 GET

The GET method is used to retrieve information about this subscription. Upon success, the response contains entity body with the data type describing the subscription.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in tables 7.15.3.1-1 and 7.15.3.1-2.

Table 7.15.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.15.3.1-2: Data structures supported by the GET request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	Data type	Cardinality	Response Codes	Remarks
	{NotificationSubscription}	1	200 OK	Upon success, a response body containing data type describing the specific distance event subscription is returned. The allowed data types for subscriptions are defined in clause 6.3.9: <ul style="list-style-type: none"> UserDistanceSubscription.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.15.3.2 PUT

The PUT method is used to update the existing subscription. PUT method in this case has "replace" semantics. Upon successful operation, the target resource is updated with new Data Type received within the message body of the PUT request.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in tables 7.15.3.2-1 and 7.15.3.2-2.

Table 7.15.3.2-1: URI query parameters supported by the PUT method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.15.3.2-2: Data structures supported by the PUT request/response on this resource

Request body	Data type	Cardinality	Remarks	
	{NotificationSubscription}	1	New NotificationSubscription is included as entity body of the request. The allowed data types for subscriptions are defined in clause 6.3.9: <ul style="list-style-type: none"> UserDistanceSubscription. 	
Response body	Data type	Cardinality	Response Codes	Remarks
	{NotificationSubscription}	1	200 OK	Upon success, a response body containing data type describing the updated subscription is returned. The allowed data types for subscriptions are defined in clause 6.3.9: <ul style="list-style-type: none"> UserDistanceSubscription.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	412 Precondition Failed	It is used when a condition has failed during conditional requests, e.g. when using Etags to avoid write conflicts when using PUT. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

	ProblemDetails	0..1	422 Unprocessable Entity	It is used to indicate that the server understands the content type of the request entity and that the syntax of the request entity is correct but that the server is unable to process the contained instructions. This error condition can occur if a JSON request body is syntactically correct but semantically incorrect, for example if the target area for the request is considered too large. This error condition can also occur if the capabilities required by the request are not supported. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.15.3.3 PATCH

Not applicable.

7.15.3.4 POST

Not applicable.

7.15.3.5 DELETE

The DELETE method is used to cancel the existing subscription. Cancellation can be made by deleting the resource that represents existing subscription.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in tables 7.15.3.5-1 and 7.15.3.5-2.

Table 7.15.3.5-1: URI query parameters supported by the DELETE method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.15.3.5-2: Data structures supported by the DELETE request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	Data type	Cardinality	Response Codes	Remarks
	n/a		204 No Content	Upon success, a response 204 No Content without any response body is returned.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.16 Resource: area_subscriptions

7.16.1 Description

This resource contains various resources related to subscriptions for notifications related to location events.

7.16.2 Resource definition

Resource URI: {apiRoot}/location/v3/subscriptions/area

This resource shall support the resource URI variables defined in table 7.16.2-1.

Table 7.16.2-1: Resource URI variables for resource "area_subscriptions"

Name	Definition
apiRoot	See clause 7.2.

7.16.3 Resource methods

7.16.3.1 GET

The GET method is used to request information about the subscriptions for this requestor. Upon success, the response contains entity body with the list of links to the subscriptions that are present for the requestor.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in tables 7.16.3.1-1 and 7.16.3.1-2.

Table 7.16.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
subscription_type	String	0..1	Query parameter to filter on a specific subscription type. Permitted values: <ul style="list-style-type: none"> event.

Table 7.16.3.1-2: Data structures supported by the GET request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	Data type	Cardinality	Response Codes	Remarks
	NotificationSubscriptionList	1	200 OK	Upon success, a response body containing the list of links to requestor's subscriptions is returned.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.16.3.2 PUT

Not applicable.

7.16.3.3 PATCH

Not applicable.

7.16.3.4 POST

The POST method is used to create a new subscription to area notifications. Upon success, the response contains entity body describing the created subscription.

This method shall support the request and response data structures, and response codes, as specified in table 7.16.3.4-1.

Table 7.16.3.4-1: Data structures supported by the POST request/response on this resource

Request body	Data type	Cardinality	Remarks	
	{NotificationSubscription}	1	The entity body in the request contains data type of the specific UE location event subscription that is to be created, where the data type options are listed below and defined in clause 6.3.8: <ul style="list-style-type: none"> UserAreaSubscription. 	
Response body	Data type	Cardinality	Response Codes	Remarks
	{NotificationSubscription}	1	201 Created	Indicates successful resource creation, where the resource URI shall be returned in the HTTP Location header field. In the returned NotificationSubscription structure, the created subscription is described using the appropriate data type from the list below and as defined in clause 6.3.8: <ul style="list-style-type: none"> UserAreaSubscription.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	415 Unsupported Media Type	It is used to indicate that the server or the client does not support the content type of the entity body. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

	ProblemDetails	0..1	422 Unprocessable Entity	It is used to indicate that the server understands the content type of the request entity and that the syntax of the request entity is correct but that the server is unable to process the contained instructions. This error condition can occur if an JSON request body is syntactically correct but semantically incorrect, for example if the target area for the request is considered too large. This error condition can also occur if the capabilities required by the request are not supported. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.16.3.5 DELETE

Not applicable.

7.17 Resource: individual area_subscription

7.17.1 Description

This resource represents a subscription that the client has created to receive location event notifications.

7.17.2 Resource definition

Resource URI: **{apiRoot}/location/v3/subscriptions/area/{subscriptionId}**

This resource shall support the resource URI variables defined in table 7.17.2-1.

Table 7.17.2-1: Resource URI variables for resource "individual area_subscription"

Name	Definition
apiRoot	See clause 7.2.
subscriptionId	Refers to created subscription, where the Location API allocates a unique resource name for this subscription. The resource name can be also used to identify the resource.

7.17.3 Resource methods

7.17.3.1 GET

The GET method is used to retrieve information about this subscription. Upon success, the response contains entity body with the data type describing the subscription.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in tables 7.17.3.1-1 and 7.17.3.1-2.

Table 7.17.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.17.3.1-2: Data structures supported by the GET request/response on this resource

Request body	Data type	Cardinality	Remarks	
n/a				
Response body	Data type	Cardinality	Response Codes	Remarks
	{NotificationSubscription}	1	200 OK	Upon success, a response body containing data type describing the specific zone subscription is returned. The allowed data types for subscriptions are defined in clause 6.3.8: <ul style="list-style-type: none"> UserAreaSubscription.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.17.3.2 PUT

The PUT method is used to update the existing subscription. PUT method in this case has "replace" semantics. Upon successful operation, the target resource is updated with new Data Type received within the message body of the PUT request.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in tables 7.17.3.2-1 and 7.17.3.2-2.

Table 7.17.3.2-1: URI query parameters supported by the PUT method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.17.3.2-2: Data structures supported by the PUT request/response on this resource

Request body	Data type	Cardinality	Remarks	
	{NotificationSubscription}	1	New NotificationSubscription is included as entity body of the request. The allowed data types for subscriptions are defined in clause 6.3.8: <ul style="list-style-type: none"> UserAreaSubscription. 	
Response body	Data type	Cardinality	Response Codes	Remarks
	{NotificationSubscription}	1	200 OK	Upon success, a response body containing data type describing the updated subscription is returned. The allowed data types for subscriptions are defined in clause 6.3.8: <ul style="list-style-type: none"> UserAreaSubscription.
	ProblemDetails	0..1	400 Bad Request	It is used to indicate that incorrect parameters were passed to the request. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	406 Not Acceptable	It is used to indicate that the server cannot provide the any of the content formats supported by the client. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	412 Precondition Failed	It is used when a condition has failed during conditional requests, e.g. when using Etags to avoid write conflicts when using PUT. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

	ProblemDetails	0..1	422 Unprocessable Entity	It is used to indicate that the server understands the content type of the request entity and that the syntax of the request entity is correct but that the server is unable to process the contained instructions. This error condition can occur if a JSON request body is syntactically correct but semantically incorrect, for example if the target area for the request is considered too large. This error condition can also occur if the capabilities required by the request are not supported. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

7.17.3.3 PATCH

Not applicable.

7.17.3.4 POST

Not applicable.

7.17.3.5 DELETE

The DELETE method is used to cancel the existing subscription. Cancellation can be made by deleting the resource that represents existing subscription.

This method shall support the URI query parameters, request and response data structures, and response codes, as specified in tables 7.17.3.5-1 and 7.17.3.5-2.

Table 7.17.3.5-1: URI query parameters supported by the DELETE method on this resource

Name	Data type	Cardinality	Remarks
n/a			

Table 7.17.3.5-2: Data structures supported by the DELETE request/response on this resource

Request body	Data type	Cardinality	Remarks	
	n/a			
Response body	Data type	Cardinality	Response Codes	Remarks
	n/a		204 No Content	Upon success, a response 204 No Content without any response body is returned.
	ProblemDetails	0..1	401 Unauthorized	It is used when the client did not submit credentials. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	1	403 Forbidden	The operation is not allowed given the current status of the resource. More information shall be provided in the "detail" attribute of the "ProblemDetails" structure.
	ProblemDetails	0..1	404 Not Found	It is used when a client provided a URI that cannot be mapped to a valid resource URI. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.
	ProblemDetails	0..1	429 Too Many Requests	It is used when a rate limiter has triggered. In the returned ProblemDetails structure, the "detail" attribute should convey more information about the error.

Annex A (informative): Complementary material for API utilization

To complement the definitions for each method and resource defined in the interface clauses of the present document, ETSI MEC ISG is providing for the Location API a supplementary description file compliant to the OpenAPI Specification [i.3].

In case of discrepancies between the supplementary description file and the related data structure definitions in the present document, the data structure definitions take precedence.

The supplementary description file, relating to the present document, is located at https://forge.etsi.org/rep/gitweb.cgi/MEC.GS_013.git.

Annex B (informative): Examples for the usage of Location API

B.1 Examples for UE Location Lookup

B.1.1 Example: Retrieve UE location by the address of a user

In this example, the client issues a request to fetch the user location of a certain user identified by its IP address.

Request

```
GET /exampleAPI/location/v3/queries/users?address=acr%3A10.0.0.1 HTTP/1.1
Host: example.com
Accept: application/json
```

Response

```
HTTP/1.1 200 OK
Date: Sun, 01 Jan 2017 00:38:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"userList": {
  "user": [
    {
      "address": "acr:10.0.0.1",
      "accessPointId": "00101000000000000000000000000001",
      "zoneId": "zone01",
      "resourceURL":
"http://example.com/exampleAPI/location/v3/queries/users?address=acr%3A10.0.0.1",
      "timeStamp": {
        "seconds": 1483231138,
        "nanoSeconds": 0
      },
      "locationInfo": {
        "latitude": 80.123,
        "longitude": 70.123,
        "altitude": 10.0,
        "accuracy": 0
      }
    }
  ],
  "resourceURL": "http://example.com/exampleAPI/location/v3/queries/users?address=acr%3A10.0.0.1"
}}
```

B.1.2 Example: Retrieve UE Location for the users in a zone

In this example, the client issues a request to retrieve a list of all the users connected to all the access points associated to the zone specified in the request.

Request

```
GET /exampleAPI/location/v3/queries/users?zoneId=zone01 HTTP/1.1
Host: example.com
Accept: application/json
```

Response

```
HTTP/1.1 200 OK
Date: Sun, 01 Jan 2017 00:38:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"userList": {
  "user": [
    {
      "address": "acr:10.0.0.1",
```

```

    "accessPointId": "00101000000000000000000000000001",
    "zoneId": "zone01",
    "resourceURL":
"http://example.com/exampleAPI/location/v3/queries/users?address=acr%3A10.0.0.1",
    "timeStamp": {
      "seconds": 1483231138,
      "nanoSeconds": 0
    }
  },
  {
    "address": "acr:10.0.0.2",
    "accessPointId": "00101000000000000000000000000001",
    "zoneId": "zone01",
    "resourceURL":
"http://example.com/exampleAPI/location/v3/queries/users?address=acr%3A10.0.0.2",
    "timeStamp": {
      "seconds": 1483231138,
      "nanoSeconds": 0
    }
  },
  {
    "address": "acr:10.0.0.3",
    "accessPointId": "00101000000000000000000000000002",
    "zoneId": "zone01",
    "resourceURL":
"http://example.com/exampleAPI/location/v3/queries/users?address=acr%3A10.0.0.3",
    "timeStamp": {
      "seconds": 1483231138,
      "nanoSeconds": 0
    }
  }
],
"resourceURL": "http://example.com/exampleAPI/location/v3/queries/users?zoneId=zone01"
}}

```

B.2 Examples for UE Location Subscribe

B.2.1 Example: create a UE Location event subscription

In this example, the client issues a request to create a UE Location event subscription for a certain user identified by its IP address. The subscription is meant to receive a notification about the "ENTERING_AREA_EVENT" user event, i.e. when the requested UE enters an area which is comprised of one or more access points. If no "locationEventCriteria" is included in the request, then the subscription is created for all the events. If no "accessPointList" is included in the request, then the subscription is created for all the events happened at all the access points.

Request

```

POST /exampleAPI/location/v3/subscriptions/users HTTP/1.1
Content-Type: application/json
Content-Length: nnnn
Accept: application/json
Host: example.com

```

```

{"userLocationEventSubscription": {
  "subscriptionType": "UserLocationEventSubscription",
  "clientCorrelator": "0123",
  "callbackReference": "http://clientApp.example.com/location_notifications/123456",
  "address": "acr:10.0.0.1",
  "locationEventCriteria": ["ENTERING_AREA_EVENT"]
}}

```

Response

```

HTTP/1.1 201 Created
Content-Type: application/json
Location : http://example.com/exampleAPI/location/v3/subscriptions/users/subscription123
Content-Length: nnnn
Date: Sun, 01 Jan 2017 00:38:59 GMT

```

```

{"userLocationEventSubscription": {
  "subscriptionType": "UserLocationEventSubscription",
  "clientCorrelator": "0123",

```

```

  "_links": {
    Self: "http://example.com/exampleAPI/location/v3/subscriptions/users/subscription123"
  },
  "callbackReference": "http://clientApp.example.com/location_notifications/123456",
  "address": "acr:10.0.0.1",
  "locationEventCriteria": ["ENTERING_AREA_EVENT"]
}}

```

B.2.2 Example: client notification about UE entering an area

In this example, the client receives a notification of a location change of the requested UE, identified by its IP address. The notification relates to the "ENTERING_AREA_EVENT" user event, i.e. when the requested UE enters one access point.

Request

```

POST /location_notifications/123456 HTTP/1.1
Content-Type : application/json
Content-Length: nnnn
Accept: application/json
Host: clientApp.example.com

{"userLocationEventNotification": {
  "notificationType": "UserLocationEventNotification",
  "timeStamp": {
    "seconds": 1483231138,
    "nanoSeconds": 0},
  "address": "acr:10.0.0.1",
  "userLocationEvent": "ENTERING_AREA_EVENT",
  "zoneId" : "zone01",
  "accessPointId" : "00101000000000000000000000000002",
  "_links" : {
    "subscription" :
"http://example.com/exampleAPI/location/v3/subscriptions/users/subscription123"
  }}}

```

Response

```

HTTP/1.1 204 No Content
Date: Sun, 01 Jan 2017 00:38:59 GMT

```

B.2.3 Example: create a UE Location periodic subscription

In this example, the client issues a request to create a UE Location subscription for a certain UE. The client expects to receive periodic notifications about the location of the specified UE at the time interval specified in the request.

Request

```

POST /exampleAPI/location/v3/subscriptions/users HTTP/1.1
Content-Type: application/json
Content-Length: nnnn
Accept: application/json
Host: example.com

{"userLocationPeriodicSubscription": {
  "subscriptionType": "UserLocationPeriodicSubscription",
  "address": "acr:10.0.0.1",
  "callbackReference": "http://clientApp.example.com/location_notifications/123456",
  "clientCorrelator": "0123",
  "periodicEventInfo": {
    "reportingAmount": 10,
    "reportingInterval": 100
  }
}}

```

Response

```

HTTP/1.1 201 Created
Content-Type : application/json
Location : http://example.com/exampleAPI/location/v3/subscriptions/users/subscription123
Content-Length: nnnn
Date: Sun, 01 Jan 2017 00:38:59 GMT

{"userLocationPeriodicSubscription": {
  "subscriptionType": "UserLocationPeriodicSubscription",
  "address": "acr:10.0.0.1",
  "callbackReference": "http://clientApp.example.com/location_notifications/123456",
  "clientCorrelator": "0123",
  "_links": {
    "self": "http://example.com/exampleAPI/location/v3/subscriptions/users/subscription123"
  },
  "periodicEventInfo": {
    "reportingAmount": 10,
    "reportingInterval": 100
  }
}}

```

B.2.4 Example: client notification about UE location in a fixed interval

Request

```

POST /location_notifications/123456 HTTP/1.1
Content-Type : application/json
Content-Length : nnnn
Accept : application/json
Host : clientApp.example.com

{"userLocationPeriodicNotification" : {
  "notificationType" : "UserLocationPeriodicNotification",
  "timeStamp" : {
    "seconds" : 1483231138,
    "nanoSeconds" : 0},
  "address": "acr:10.0.0.1",
  "result": "SUCCESS",
  "zoneId": "zone01",
  "accessPointId" : "0010100000000000000000000000000002",
  "locationInfo" : {
    "latitude" : 80.123,
    "longitude": 70.123,
    "altitude": 10.0,
    "accuracy": 0
  }
  "_links": {
    "subscription":
"http://example.com/exampleAPI/location/v3/subscriptions/users/subscription123"
  }}}

```

Response

```

HTTP/1.1 204 No Content
Date: Sun, 01 Jan 2017 00:38:59 GMT

```

B.3 Examples for Zone List Lookup and Access Point List Lookup

B.3.1 Example: Retrieve zone list

In this example, the client issues a request to fetch all information about zones related to the MEC system. After that, the client might enquire about user information or subscribe to user location by zone Id. How to define the relationship between the zones and the MEC system is out of the scope of the present document.

Request

```
GET /exampleAPI/location/v3/queries/zones HTTP/1.1
Host: example.com
Accept: application/json
```

Response

```
HTTP/1.1 200 OK
Date: Sun, 01 Jan 2017 00:38:59 GMT
Content-Type : application/json
Content-Length : nnnn
```

```
{ "zoneList" : {
  "zone" : [
    {
      "zoneId" : "zone01",
      "numberOfAccessPoints" : 3,
      "numberOfUnserviceableAccessPoints" : 1,
      "numberOfUsers" : 10,
      "resourceURL" : "http://example.com/exampleAPI/location/v3/queries/zones/zone01"
    },
    {
      "zoneId" : "zone02",
      "numberOfAccessPoints" : 12,
      "numberOfUnserviceableAccessPoints" : 0,
      "numberOfUsers" : 36,
      "resourceURL" : "http://example.com/exampleAPI/location/v3/queries/zones/zone02"
    }
  ],
  "resourceURL" : "http://example.com/exampleAPI/location/v3/queries/zones "
}
}
```

B.3.2 Example: Retrieve Access Point List

In this example, the client issues a request to fetch all information about access points related to the zone. How to define the relationship between the zone and the access points is out of the scope of the present document.

Request

```
GET /exampleAPI/location/v3/queries/zones/zone01/accessPoints HTTP/1.1
Host: example.com
Accept: application/json
```

Response

```
HTTP/1.1 200 OK
Date: Sun, 01 Jan 2017 00:38:59 GMT
Content-Type : application/json
Content-Length : nnnn
```

```
{ "accessPointList" : {
  "zoneId" : "zone01",
  "accessPoint" : [
    {
      "accessPointId" : "ap01",
      "locationInfo" : {
        "latitude" : 90.123,
        "longitude" : 80.123,
        "accuracy" : 0
      },
      "connectionType" : "5G NR",
      "operationStatus" : "Serviceable",
      "numberOfUsers" : 5,
      "resourceURL" : "http://example.com/exampleAPI/location/v3/queries/zones/zone01/accessPoints/ap01"
    },
    {
      "accessPointId" : "ap02",
      "locationInfo" : {
        "latitude" : 93.123,
        "longitude" : 83.123,
        "accuracy" : 3
      }
    }
  ]
}
```

```

    },
    "connectionType": "5G NR",
    "operationStatus": "Serviceable",
    "numberOfUsers": 20,
    "resourceURL": "http://example.com/exampleAPI/location/v3/queries/zones/zone01/accessP
oints/ap02"
  }
],
"resourceURL" : "http ://example.com/exampleAPI/location/v3/queries/zones/zone01/accessPoints
}
}

```

B.4 Examples for Zone Location Event Subscribe and Zone Status Subscribe

B.4.1 Example: create a zone location subscription

In this example, the client issues a request to create a zone location event subscription for a zone. The subscription is meant to receive a notification about the "ENTERING_AREA_EVENT" user event, i.e. when any UE enters this zone. If no user event is included in the request, then the subscription is created for all the events.

Request

```

POST /exampleAPI/location/v3/subscriptions/zones HTTP/1.1
Content-Type : application/json
Content-Length : nnnn
Accept : application/json
Host : example.com

{"zoneLocationEventSubscription" : {
  "subscriptionType" : "ZoneLocationEventSubscription",
  "clientCorrelator" : "0123",
  "callbackReference" : "http://clientApp.example.com/location_notifications/123456",
  "zoneId": "zone01",
  "locationEventCriteria": ["ENTERING_AREA_EVENT"]
}}

```

Response

```

HTTP/1.1 201 Created
Content-Type: application/json
Location : http://example.com/exampleAPI/location/v3/subscriptions/zones/subscription123
Content-Length: nnnn

Date: Sun, 01 Jan 2017 00:38:59 GMT
{"zoneLocationEventSubscription": {
  "subscriptionType": "ZoneLocationEventSubscription",
  "clientCorrelator": "0123",
  "_links": {
    "self": "http://example.com/exampleAPI/location/v3/subscriptions/zones/subscription123"},
    "callbackReference": "http://clientApp.example.com/location_notifications/123456",
    "zoneId": "zone01",
    "locationEventCriteria": ["ENTERING_AREA_EVENT"]
  }}

```

B.4.2 Example: client notification about UE entering the zone

In this example, the client receives a notification about a certain UE enters the zone subscribed by the client. The notification relates to the "ENTERING_AREA_EVENT" user event.

Request

```

POST /location_notifications/123456 HTTP/1.1
Content-Type : application/json
Content-Length : nnnn
Accept : application/json
Host : clientApp.example.com

```



```
{
  "zoneLocationEventNotification" : {
    "notificationType" : "ZoneLocationEventNotification",
    "address" : "acr :10.0.0.2",
    "zoneId" : "zone01",
    "userLocationEvent" : "ENTERING_AREA_EVENT",
    "timeStamp": {
      "seconds": 1483231138,
      "nanoSeconds": 0},
    "_links": {
      "subscription": "http://example.com/exampleAPI/location/v3/subscriptions/zones/subscription123"
    }
  }
}
```

Response

```
HTTP/1.1 204 No Content
Date: Sun, 01 Jan 2017 00:38:59 GMT
```

B.4.3 Example: create a zone status subscription

In this example, the client issues a request to create a zone status subscription for a zone. The client expects to be notified by the server when the user number of this zone reaches a threshold set in the subscribe request message.

Request

```
POST /exampleAPI/location/v3/subscriptions/zones HTTP/1.1
Content-Type : application/json
Content-Length : nnnn
Accept : application/json
Host : example.com
```

```
{
  "zoneStatusSubscription" : {
    "subscriptionType" : "ZoneStatusSubscription",
    "clientCorrelator" : "0123",
    "callbackReference" : "http://clientApp.example.com/location_notifications/123456",
    "zoneId": "zone01",
    "upperNumberOfUsersZoneThreshold": 1000,
    "lowerNumberOfUsersZoneThreshold": 10
  }
}
```

Response

```
HTTP/1.1 201 Created
Content-Type : application/json
Location : http://example.com/exampleAPI/location/v3/subscriptions/zones/subscription123
Content-Length: nnnn

Date: Sun, 01 Jan 2017 00:38:59 GMT
{"zoneStatusSubscription": {
  "subscriptionType": "ZoneStatusSubscription",
  "clientCorrelator": "0123",
  "_links": {
    "self": "http://example.com/exampleAPI/location/v3/subscriptions/zones/subscription123"},
  "callbackReference": "http://clientApp.example.com/location_notifications/123456",
  "zoneId": "zone01"
}}
```

B.4.4 Example: client notification about UE number reaching the threshold

In this example, the client receives a notification about the UE number of the zone reaches the upper threshold that has already been set in the subscription.

Request

```
POST /location_notifications/123456 HTTP/1.1
Content-Type : application/json
Content-Length : nnnn
Accept : application/json
```

```
Host : clientApp.example.com

{"zoneStatusNotification" : {
  "notificationType" : "ZoneStatusNotification",
  "userNumEvent" : "OVER_ZONE_UPPER_THD",
  "zoneId" : "zone01",
  "timeStamp": {
    "seconds": 1483231138,
    "nanoSeconds": 0},
  "_links": {
    "subscription": "http://example.com/exampleAPI/location/v3/subscriptions/zones/subscription123"
  }}}}
```

Response

```
HTTP/1.1 204 No Content
Date: Sun, 01 Jan 2017 00:38:59 GMT
```

B.5 Examples for UE Distance Lookup

B.5.1 Example: retrieve distance between two UEs

In this example, the client issues a request to fetch the distance between two UEs identified by their IP addresses.

Request

```
GET /exampleAPI/location/v3/queries/distance?address=acr%3A10.0.0.1&address=acr%3A10.0.0.2 HTTP/1.1
Content-Type: application/json
Content-Length: nnnn
Accept: application/json
Host: example.com
```

Response

```
HTTP/1.1 201 Created
Date: Sun, 01 Jan 2020 00:38:59 GMT
Content-Type: application/json
Content-Length: nnnn
{"terminalDistance": {
  "distance": 1000
}}
```

B.6 Examples for UE Distance Subscribe

B.6.1 Example: create a UE Distance event subscription

In this example, the client issues a request to create UE distance event subscription. The subscription is meant to receive a notification about the "AllWithinDistance" distance event, i.e. when all monitored UEs are within a specified distance.

Request

```
POST /exampleAPI/location/v3/subscriptions/distance HTTP/1.1
Content-Type: application/json
Content-Length: nnnn
Accept: application/json
Host: example.com

{"userDistanceSubscription": {
  "subscriptionType": "UserDistanceSubscription",
  "clientCorrelator": "0123",
  "callbackReference": "http://clientApp.example.com/location_notifications/123456",
  "monitoredAddress": ["acr:10.0.0.1", "acr:10.0.0.2", "acr:10.0.0.3"],
  "distance": 100.0,
  "trackingAccuracy": 10.0,
  "criteria": "AllWithinDistance",
  "checkImmediate": false
}}
```

```
}}
```

Response

```
HTTP/1.1 201 Created
Content-Type: application/json
Location : http://example.com/exampleAPI/location/v3/subscriptions/distance/subscription123
Content-Length: nnnn

Date: Sun, 01 Jan 2017 00:38:59 GMT
{"userDistanceSubscription": {
  "subscriptionType": "UserDistanceSubscription",
  "clientCorrelator": "0123",
  "_links": {
    "self": "http://example.com/exampleAPI/location/v3/subscriptions/distance/subscription123"},
  "callbackReference": "http://clientApp.example.com/location_notifications/123456",
  "monitoredAddress": ["acr:10.0.0.1", "acr:10.0.0.2", "acr:10.0.0.3"],
  "distance": 100.0,
  "trackingAccuracy": 10.0,
  "criteria": "AllWithinDistance",
  "checkImmediate": false
}}
```

B.6.2 Example: client notification about all monitored devices within the specified distance

In this example, the client receives a notification about all monitored UEs are within the specified distance. The notification relates to the "AllWithinDistance" user event.

Request

```
POST /location_notifications/123456 HTTP/1.1
Content-Type: application/json
Content-Length: nnnn
Accept: application/json
Host: clientApp.example.com

{"userDistanceNotification": {
  "notificationType": "UserDistanceNotification",
  "monitoredUsers": {
    "user": [
      {
        "address": "acr:10.0.0.1",
        "accessPointId": "00101000000000000000000000000001",
        "zoneId": "zone01",
        "timeStamp": {
          "seconds": 1483231138,
          "nanoSeconds": 0
        },
        "locationInfo": {
          "latitude": 80.123,
          "longitude": 70.123,
          "altitude": 10.0,
          "accuracy": 10
        }
      },
      {
        "address": "acr:10.0.0.2",
        "accessPointId": "00101000000000000000000000000001",
        "zoneId": "zone01",
        "timeStamp": {
          "seconds": 1483231138,
          "nanoSeconds": 0
        },
        "locationInfo": {
          "latitude": 81.372,
          "longitude": 71.463,
          "altitude": 10.0,
          "accuracy": 10
        }
      },
      {
        "address": "acr:10.0.0.3",
        "accessPointId": "00101000000000000000000000000001",
```

```

        "zoneId" : "zone01",
        "timeStamp" : {
            "seconds" : 1483231138,
            "nanoSeconds": 0
        },
        "locationInfo": {
            "latitude": 82.511,
            "longitude": 72.804,
            "altitude": 10.0,
            "accuracy": 10
        }
    }
}
],
},
"distanceEvent": "AllWithinDistance",
"_links": {
"subscription":http://example.com/exampleAPI/location/v3/subscriptions/distance/subscription123
}
}}

```

Response

```

HTTP/1.1 204 No Content
Date: Sun, 01 Jan 2017 00:38:59 GMT

```

B.7 Examples for UE Area Subscribe

B.7.1 Example: Create a UE Area subscription

In this example, the client issues a request to create a UE Area subscription for a certain user identified by its IP address. The client expects to receive a notification when the requested UE enters the specified circle.

Request

```

POST /exampleAPI/location/v3/subscriptions/area HTTP/1.1
Content-Type: application/json
Content-Length: nnnn
Accept: application/json
Host: example.com

```

```

{"userAreaSubscription": {
  "subscriptionType": "UserAreaSubscription",
  "clientCorrelator": "0123",
  "callbackReference": "http://clientApp.example.com/location_notifications/123456",
  "addressList": ["acr:10.0.0.1"],
  "trackingAccuracy": 10.0,
  "locationEventCriteria": ["ENTERING_AREA_EVENT"],
  "areaDefine": {
    "shape": 1,
    "points": [{
      "latitude": -80.86302,
      "longitude": 41.277306
    }],
    "radius": 500
  }
}
}}

```

Response

```

HTTP/1.1 201 Created
Content-Type : application/json
Location : http://example.com/exampleAPI/location/v3/subscriptions/area/subscription123
Content-Length: nnnn

```

```

Date: Sun, 01 Jan 2017 00:38:59 GMT
{"userAreaSubscription": {
  "subscriptionType": "UserAreaSubscription",
  "clientCorrelator": "0123",
  "callbackReference": "http://clientApp.example.com/location_notifications/123456",
  "addressList": ["acr:10.0.0.1"],
  "trackingAccuracy": 10.0,
  "locationEventCriteria": ["ENTERING_AREA_EVENT"],

```

```
"areaDefine": {
  "shape": 1,
  "points": [{
    "latitude": -80.86302,
    "longitude": 41.277306
  }],
  "radius": 500
},
"_links": {
  Self: "http://example.com/exampleAPI/location/v3/subscriptions/area/subscription123"
}
}}
```

B.7.2 Example: Client notification about UE Area change

In this example, the client receives a notification of the location of the requested UE, identified by its IP address.

Request

```
POST /location_notifications/123456 HTTP/1.1
Content-Type : application/json
Content-Length: nnnn
Accept: application/json
Host: clientApp.example.com
```

```
{"userAreaNotification": {
  "notificationType": "UserAreaNotification",
  "addressList": ["acr:10.0.0.1"],
  "userLocationEvent": "ENTERING_AREA_EVENT",
  "_links": {
    Self: "http://example.com/exampleAPI/location/v3/subscriptions/area/subscription123"
  }
}
}}
```

Response

```
HTTP/1.1 204 No Content
Date: Sun, 01 Jan 2017 00:38:59 GMT
```

History

Document history		
V1.1.1	July 2017	Publication
V2.1.1	September 2019	Publication
V2.2.1	January 2022	Publication
V3.1.1	January 2023	Publication
V3.2.1	November 2024	Publication