ERICSSON

Service Exposure OneAPI Location Development

Ericsson Composition Engine

USER GUIDE

44/1553-CXP 904 0189 Uen H

Contents

1 About this Document

This chapter contains the following parts:

- Revision information
- Purpose
- Target audience
- Prerequisites

1.1 Revision Information

The following table shows the changes in recent revisions. Other than editorial changes, this document has been revised as follows:

Table 1 Revision Information

Revision	Section	Change
D	The whole document	Changed the resource URI.
E	Section 3 on page 5	Added description of Authorization header.
F	Section 5.1.3 on page 10	Updated the description of `altitude`.
	Section 6.2.3 on page 16	Clarified connection between the error codes, service exception `SVC0001`, and policy exception `POL0001`.
	Section 6.2.3 on page 16	Added error codes 9075 and 9076.
	Section 6.2.1 on page 14	Added `SVC0200` exception.
	Section 6.2.3 on page 16	Added error code `62`.
G	Section 3 on page 5	Added information about the Access Key header in B2B scenario.
	Section 3 on page 5	Corrected authentication parameters in Authentication Parameters.
	Section 6.2.3 on page 16	Added reference to the integration adapter error codes in *Ericsson Composition Engine, System Administration Guide*, Reference [3].
H	Section 3 on page 5	Updated the description about OAuth authentication.

1.2 Purpose

This document describes how to develop an application which is compatible with Service Exposure OneAPI Location interface. The full specification of

OneAPI Location interface and examples with detailed explanation are provided in this document.

1.3 Target Group

This document is intended for application developers.

1.4 Prerequisites

It is assumed that the reader of this document is familiar with the following knowledge:

- Parlay REST API
- HTTP protocol
- RESTful HTTP
- ECE Service Exposure structure

2 OneAPI Location Overview

The OneAPI Location interface allows an application to query the location of one or more mobile devices. These mobile devices are connected to a mobile operator network.

OneAPI is a profile (subset) of **Parlay REST v1.0**. The full specifications and guidelines are available at:
ttp://www.openmobilealliance.org/Technical/current_releases.aspx

For more information or examples about **Location RESTful NetAPI**, you can visit:
http://oneapi.gsma.com/Location-restful-netapi/.

The supportability of the Location RESTful NetAPI operations in OneAPI Location service are listed in the following table:

Table 2 Supported Operations in OneAPI Location

Resource URI	HTTP Method	Operation	Supportability in OneAPI
http://example.com/oneapi/location/1/queries/location?address={address}&requestedAccuracy={metres}	GET	Request the location of one or more mobile terminals	Yes

Note: It might be different on the supported operations between OneAPI and the RESTful NetAPI specification. For detailed operation supportability in OneAPI Location service, see the corresponding section in the following chapters.

3 Security and Authentication

A server-side certificate is used to secure the HTTPS connection and to confirm the server identity. The client is authenticated with HTTP basic authentication or with OAuth authentication.

HTTP Basic Authentication

When the client uses HTTP basic authentication, add an Authorization header in the request as follows:

```
GET /oneapi/location/1/queries/location?address=tel%3A%2B96050108714&
requestedAccuracy=100 HTTP/1.1
Connection: close
Authorization: Basic cGFydG5lcjFAYXBwMTphdXRob2s=
Accept: application/json
Host: 10.175.163.203:38080
User-Agent: Apache-HttpClient/4.2.1 (java 1.5)
```

The format of the Authorization header is as follows:

`Authorization: Basic` *<base64 encoded (user-pass)>*

The following parameters are used in the authentication.

Table 3 Authentication Parameters

Parameter	Value
`user-pass`	`userid:password`
`userid`	*<application-id>@<partner-id>*
`password`	`TEXT`

For example, if the `userid` is `app1@partner1` and the `password` is `authok`, then the header including the base64 encoded `user-pass` is the following:
`Authorization: Basic cGFydG5lcjFAYXBwMTphdXRob2s=`

For more information, see *Ericsson Composition Engine, System Administration Guide*, Reference [3].

OAuth Authentication

For OAuth authentication, only the client credential flow is supported for the client to get the access token. The OAuth authorization flow which uses authorization code to get access token is not supported. For more information about client integration, see *Authorization Integration Guide*, Reference [1].

When the client uses OAuth authentication, add an Authorization header in the request as follows:

```
GET /oneapi/location/1/queries/location?address=tel%3A%2B96050108714
&requestedAccuracy=100 HTTP/1.1
Connection: close
Authorization: Bearer MkdPhwf2FCVm5rijcaCi
Accept: application/json
Host: 10.175.163.203:38080
User-Agent: Apache-HttpClient/4.2.1 (java 1.5)
```

The Authorization header contains `Bearer`, followed by an identifier from the OAuth server.

Access Key

If the application identifier is a globally unique access key in a Business to Business (B2B) scenario, add an Access Key header after the Authorization header in the request. The Access Key header contains the application access key as follows:

`AccessKey:` *<access key of application>*

Following is an example of the request containing the Access Key header:

```
GET /oneapi/location/1/queries/location?address=tel%3A%2B96050108714
&requestedAccuracy=100 HTTP/1.1
Connection: close
Authorization: Bearer MkdPhwf2FCVm5rijcaCi
AccessKey: 1316kj
Accept: application/json
Host: 10.175.163.203:38080
User-Agent: Apache-HttpClient/4.2.1 (java 1.5)
```

4 Resources and URIs

OneAPI Location is a RESTful API. A RESTful API use HTTP commands `POST`, `GET`, `PUT`, and `DELETE` to operate on a resource at the server. The resource is addressed by a URI. What is returned by the server is a representation of that resource depending on its current state. HTTP `GET` command is used in OneAPI Location.

Representation Formats

The Location API supports `application/json` content type.

5 Interface: Query Location

This interface defines following operations:

- Query the location of one mobile terminal
- Query the location of multiple mobile terminals

5.1 Operation: Query Location of One Mobile Terminal

Web application can use the Query Location interface to query the location of one mobile device.

5.1.1 Resource and URI

Table 4 Resource and URI

Resource URI	HTTP Method	Operation
https://*<serverRoot>*/oneapi/location/1/queries/location?address=*<address>*&requestedAccuracy=*<metres>* (1)	GET	Query the location of one mobile terminal.

(1) <serverRoot> is replaced by the hostname of the OneAPI server that is being accessed.

5.1.2 Request

The following is an example of the request to query the location of one mobile terminal.

```
GET /oneapi/location/1/queries/location?address=tel%3A%2B4605010759
&requestedAccuracy=100 HTTP/1.1
Host: ece.example.com:38080
Accept: application/json
```

The following table shows the parameters for the request to query the location of one mobile terminal.

Table 5 Request Parameters for Querying Location of One Mobile Terminal

Parameter	Location	Type	Description	Mandatory
address	URL	String	The MSISDN in tel URI(RFC3966) format of the mobile device to locate. At least one address must be provided. Repeat the address parameter for multiple devices. "tel:" scheme and "+" identifier must be used for address, and must be URL-escaped. In the above example, the recipients MSISDN is in tel URI(RFC3966) format.%3A represents ":" and %2B represents "+". Thus tel%3A%2B4605010759 represents tel:+4605010759. The Maximum number of address is limited by SLA MaxNumberOfAddresses.	Yes
requestedAccuracy	URL	Integer	It is the preferred accuracy of the result, in metres. Generally, it takes longer time to retrieve an accurate location than a coarse location . So requestedAccuracy=10 takes longer than requestedAccuracy=100 . The highest request accuracy is limited by SLA MinGetLocationRequestedAccuracy.	Yes

5.1.3 Response

The following is an example of the response to querying the location of one mobile terminal:

```
HTTP/1.1 200 OK
Content-Type: application/json
Date: Tue, 04 Sep 2012 08:05:03 GMT

{"terminalLocationList":{
      "terminalLocation":{
                "address":"tel:+4605010759",
                "locationRetrievalStatus":"Retrieved",
                "currentLocation":{
                                  "accuracy":"100",
                                  "altitude":"+50.0",
                                  "latitude":"+59.2294",
                                  "longitude":"+018.0803",
                                  "timestamp":"2012-09-04T08:05:07Z"
                 }
}}}
```

The following table shows the response parameters.

Table 6 Response Parameters for Querying Location of One Mobile Terminal

Parameter	Description
address	The terminal located, as per RFC 3966. Only international number issupported.

Parameter	Description
accuracy	The result accuracy in metres.
altitude	The altitude of location in meters. If enabler respond with no value to Service Exposure, SE sets the altitude parameter to 0 and then send it to partner.
latitude	The latitude of location in decimal degrees, ISO 6709.
longitude	The longitude of location in decimal degrees, ISO 6709.
timestamp	In xsd:dateTime format.
locationRetrievalStatus	Location retrieval status, with following possible values: • Retrieved: retrieve location successfully. • NotRetrieved: fail to retrieve location. • Error: error in retrieving location.

5.2 Operation: Query Location of Multiple Mobile Terminals

Web application can use the Query Location interface to query the location of multiple mobile terminals.

5.2.1 Resource and URI

Table 7 Resource and URI

Resource URI	HTTP Method	Operation
https://<*serverRoot*>/oneapi/location/1/queries/location?address=<*address*>&requestedAccuracy=<*metres*>(1)(2)	GET	Query the location of multiple mobile terminals.

(1) <address> is more than one.

(2) <serverRoot> is replaced by the hostname of the OneAPI server that is being accessed.

5.2.2 Request

The following is an example of the request to query the location of multiple mobile terminals.

```
GET /oneapi/location/1/queries/location?address=tel%3A%2B9605010622&address=
tel%3A%2B9605010623&address=tel%3A%2B9605010624&requestedAccuracy=1000 HTTP/1.1
Host: ece.example.com:38080
Accept: application/json
```

The request parameters are same to the parameters shown in Table 5.

5.2.3 Response

The following is an example of the response for querying the location of multiple mobile terminals:

```
HTTP/1.1 200 OK
Content-Type: application/json
Date: Tue, 04 Sep 2012 08:05:08 GMT
{"terminalLocationList":{
        "terminalLocation":[
               {"address":"tel:+9605010000",
                    "locationRetrievalStatus":"Retrieved",
                    "currentLocation":{
                                        "accuracy":"1000",
                                        "altitude":"+50.0",
                                        "latitude":"+59.2294",
                                        "longitude":"+018.0803",
                                        "timestamp":"2012-09-04T08:05:09Z"}},
               {"address":"tel:+9605010002",
                    "locationRetrievalStatus":"Retrieved",
                    "currentLocation":{
                                         "accuracy":"1000",
                                         "altitude":"+50.0",
                                         "latitude":"+59.2294",
                                         "longitude":"+018.0803",
                                         "timestamp":"2012-09-04T08:05:09Z"}},
               {"address":"tel:+9605010001",
                   "locationRetrievalStatus":"Retrieved",
                   "currentLocation":{
                                         "accuracy":"1000",
                                         "altitude":"+50.0",
                                         "latitude":"+59.2294",
                                         "longitude":"+018.0803",
                                         "timestamp":"2012-09-04T08:05:09Z"}}
]}}
```

The parameter `terminalLocation` is an array representing the information for each terminal located. For description of the other parameters, see Table 6.

6 Response Codes and Exceptions

This chapter describes some response codes and exceptions.

6.1 Response Codes

The following table shows some response codes and their indications.

Table 8 HTTP Response Codes

Response Codes	Indication
200	Success
201	Created. The message resource is created and is being queued for delivery.
204	No content
400	Bad request. Check the error message and correct the request syntax.
401	Authentication failure. Check the authentication requirements from OneAPI provider.
403	Forbidden; please provide authentication credentials.
404	Not found: mistake in the host or path of the service URI, or the resource is not implemented.
405	Method not supported. For example, only `GET` and not `POST` is supported for a given resource.
503	Server busy and service unavailable. Please retry the request.

6.2 Exceptions

Following is an example of exception.

```
400 Bad Request
Date:  Tue, 17 Jul 2012 09:33:49 GMT
Content-Type:  application/json

{"requestError": {
    "serviceException": {
        "messageId": "SVC0002",
        "text": " Invalid input value for message part %1",
        "variables": " tel:+016309700000"
    }
}}
```

Following table shows the meaning of the parameters in the exception example.

Table 9 Parameters Description

Parameter	Description
`400`	Error code

Parameter	Description
requestError	Exception reason. It contains serviceException and policyException. They share the same body messageId, text, and variables.
serviceException	The reason why the service can not accept the request. For example, the registrationId is incorrect.
policyException	It shows that the request syntax is valid, however an operator policy is broken. In this exception example, it is because the operator do not support the batch size requested.
messageId	The identifier of the exception.
text	The description for the exception.
variables	It indicates any specific cause of the error.

6.2.1 Service Exceptions

This section lists the available service exceptions. The following table shows error codes, the possible reasons why the exception occurred, and the possible solutions.

Table 10 Service Exceptions

ID	Exception Text	Variables	HTTP Code
SVC0001	A service error occurred. Error code is %1	**%1** – explanation of the error(1)	400 Bad Request
SVC0002	Invalid input value for message part %1	**%1** – the part of the request that is invalid	400 Bad Request
SVC0003	Invalid input value for message part %1, valid values are %2.	**%1** – message part **%2** – list of valid values	400 Bad Request
SVC0004	No valid addresses provided in message part %1. Addresses means phone numbers and some like that. So this exception is thrown if the MSISDN does not follow the correct format (For example,+441234567890). The MSISDN includes URL encoding where necessary. If the address is part of the resource URL, the status code 404 is used. Otherwise the status code 400 is used.	**%1** – message part	404 Not Found 400 Bad Request
SVC0005	Correlator %1 specified in message part %2 is a duplicate. This exception is thrown if the clientCorrelator has already been used, for example, when creating a previous resource.	**%1** – correlator **%2** – message part	409 Conflict
SVC0006	Group %1 in message part %2 is not a valid group.	**%1** – identifier for the invalid group **%2** – message part	400 Bad Request
SVC0007	Invalid charging information	None	400 Bad Request
SVC0008	Overlapped criteria %1	%1 – Message Part with the overlapped criteria	400 Bad Request

ID	Exception Text	Variables	HTTP Code
SVC1000	No resources. This exception is thrown if there are no server resources available to process the request.	None	503 Service unavailable
SVC0200	Accuracy out of limit. It also means the requested Quality of Position (QOP) cannot be provided by location server.	None	403 Forbidden

(1) For a reference to the error codes, see Section 6.2.3 on page 16.

6.2.2 Policy Exceptions

This section lists the available policy exceptions. The following table shows error codes, the possible reasons why the exception occurred, and the possible solutions.

Table 11 Policy Exceptions

ID	Exception Text	Variables	HTTP Code
POL0001	A policy error occurred. Error code is %1.	**%1** – explanation of the error(1)	403 Forbidden
POL0002	Privacy verification failed for address %1, request is refused	**%1** – address privacy verification fails for	403 Forbidden
POL0003	Too many addresses specified in message part %1.	**%1** – message part	403 Forbidden
POL0004	Unlimited notification request not supported	None	403 Forbidden
POL0005	Too many notifications requested	None	403 Forbidden
POL0006	Group specified in message part %1 not allowed.	**%1** – message part Group means an address that refers to more than one end user.	403 Forbidden
POL0007	Nested groups specified in message part %1 not allowed	**%1** – message part. Note: Group means an address which refers to more than one end user. Groups cannot contain addresses which are themselves groups.	403 Forbidden
POL0008	Charging is not supported	None	403 Forbidden
POL0009	Invalid frequency requested	None	403 Forbidden

ID	Exception Text	Variables	HTTP Code
POL0010	Requested information unavailable as the retention time interval has expired.	None This exception is thrown if, for example, the delivery status of an old MMS is requested, which means that the server no longer maintains the resource. In case the information that has become unavailable is addressed by a resource URL, the following applies: • If the resource URL refers to a resource that has existed in the past and the server is aware of that fact, the status code 410 is used. • If the server is not aware, the status code 404 is used. • In all other cases, the status code 403 is used.	403 Forbidden 404 Not Found 410 Gone
POL0011	Media type not supported	None	403 Forbidden 406 Not Acceptable
POL0012	Too many description entries specified in message part %1	**%1** – message part	403 Forbidden
POL0013	Duplicated addresses %1	%1 – duplicated addresses	400 Bad Request
POL1009	User has not been provisioned for %1.	**%1** – name of the service	403 Forbidden
POL1010	User has been suspended from %1.	**%1** – name of the service	403 Forbidden
POL0230	Requested accuracy is not supported.	None	500 Internal Server Error

(1) For a reference to the error codes, see Section 6.2.3 on page 16.

6.2.3 Error Codes

This section gives a reference to the error codes in service exception SVC0001 in Table 10 and in policy exception POL0001 in Table 11.

Table 12 Error Code Reference

Error Code	Indication
1	Unexpected network or system error
2	Request time-out
3	The service capability is inactive
4	Cannot connect to rule engine
5	Subscription limitations are violated
6	Throttling rejected on service capability level
7	The distribution list adapter is not deployed

Error Code	Indication
8	There is no valid address in the distribution list
20	SLA limitations are violated
21	The application or the service provider is inactive or does not exist
22	Too many addresses are specified
23	The service activation number does not exist in the allowed list
24	Error while applying SLA enforcement rule
26	Throttling rejected on application or service provider level
27	All addresses are rejected by the integration adapter
28	The SLA property `DelayToleranceNoDelayAction` is violated and the violation action contains `reject`
29	The SLA property `DelayToleranceLowDelayAction` is violated and the violation action contains `reject`
30	The SLA property `DelayToleranceDelayTolerantAction` is violated and the violation action contains `reject`
31	The SLA property `MinGetLoactionRequestAccuracy` is violated and the corresponding violation action contains `reject`
40	Southbound resource-specific error
42	Request-specific error
44	Network traffic error
46	Vendor-specific error
47	GMLC error
48	The response time in the request is `NO_DELAY` but the system cache is disabled
49	The response time in the request is `NO_DELAY` but the location type is `CURRENT` or `CURRENT_LAST`
50	The location type in the request is `LAST` but the system cache is disabled
51	Congestion in GMLC
52	Not in cache
53	The session type is not `DIAL`
54	The Coordinate Reference System (CRS) in the request is not `WGS84`
55	The resource-related service enabler does not have a cache
56	The GMLC is not reachable
57	ADC response error
58	Device capabilities not found session error
61	The operation is not allowed for the user
62	License is invalid
70	The operation is not allowed in the southbound resource
71	The request sender or the group owner belongs to the service provider, for ParlayX ALM
72	The request sender is not authorized in the southbound resource
80	Too many subscriptions for a watcher or a presentity (ParlayX Presence)

Error Code	Indication
81	The subscription is expired or does not exist
9000–9499	Integration adapter service exception SVC0001 error codes. For more information, see *Ericsson Composition Engine, System Administration Guide*, Reference [3].
9500–9999	Integration adapter policy exception POL0001 error codes. For more information, see *Ericsson Composition Engine, System Administration Guide*, Reference [3].

Glossary

API
Application Programming Interface

B2B
Business to Business

ECE
Ericsson Composition Engine

HTTP
Hypertext Transfer Protocol

HTTPS
HTTP over SSL

ID
Identification

MMS
Short Message Service

OAuth
Open standard for Authorization

OneAPI
Open Network Enablers API

REST
Representational State Transfer

SLA
Service Level Agreement

URI
Uniform Resource Identifier

URL
Uniform Resource Locator

Reference List

Ericsson Documents

[1] *Authorization Integration Guide*, 2/1553-CXP 904 0266 Uen

[2] *Ericsson Composition Engine, Glossary*, 0033-HSC 901 024/1 Uen

[3] *Ericsson Composition Engine, System Administration Guide*, 1543-APR 901 0383/30 Uen