Open API Project

OneAPI v2.0 Terminal Location REST

Document Version 1.2

Document Revision History

Rev #	Date	Description
1.0	June 20 2013	Initial version, based on Aepona ASE1.0 OneAPI 2.0 TL API Guide doc v1.1d, with note added for requestedAccuracy parameter.
1.1	July 16 2013	Removed section 3.3 and references to multiple subscriber locations. Updated POL0002 to POL-002; POL-015 to POL-028.
1.2	Nov 1 2013	Updated URIs, examples, to match production values.

Copyright

Table of Contents

1 Terminal Location REST Overview

The Terminal Location interface allows an application to query the location of one subscriber at a time. You can find some examples of why you may want to do this in the use cases at http://www.oneapi.gsmworld.com.

A Terminal Location sandbox service is also provided, URI and examples for which are contained below and in Appendix A, and details on how to configure the service in a separate SandboxDataService API guide.

> **!** Throughout this document, the examples may be shown WITHOUT URL encoding for readability purposes, e.g. if the address "tel:+12345678900" is in the URL example, this should be encoded as "tel%3A%2B12345678900", where the character ":" is "%3A" and the character "+" is "%2B"

2 Authentication

A server side certificate is required plus HTTP Basic Authentication.

For more information, refer to the "Developer Access' section in the 'OneAPI v2.0 Common Information Guide'.

3 Methods

Terminal Location may be accessed via the REST API (described in this document). The following method is available:

- Query the Location of One Terminal

GET is used to retrieve the location (latitude/longitude) of one or more terminals. POST, PUT and DELETE are not used in OneAPI location.

3.1 URIs

The URI's used in the location API are as follows:

- Query Location URI - described below

https://developerportal.uscellular.com/services/getLocation/{apiVersion}/location/queries/location?address={address}&requestedAccuracy={metres}

- Query Location URI for Sandbox service – described in Appendix A

  https://developerportal.uscellular.com/services/getLocationSandbox/{apiVersion}/location/queries/location?address={address}&requestedAccuracy={metres}

! Representation formats – the response content type for the Location API is application/JSON.

The following request URL variables are common to all the URIs:

Name	Description
apiVersion	Version of the API that the client wants to use. In this case the API version is 2_0

3.2 Query the Location of One Terminal

This method allows you to query the location of a single mobile terminal. The location is determined using altitude, latitude and longitude.

3.2.1 Request

```
GET https://developerportal.uscellular.com/services/getLocation/2_0/location/queries/location?address=tel%3A%2B12345600001&requestedAccuracy=5000 HTTP/1.1

Authorization: Basic QWVwb25hVfghjfghohUWF6eHN3MjNl
Accept: application/json
```

3.2.2 Request Parameters

Table 1: Query Location Request Parameters

Parameter	Data Type	Description	Optional

address	xsd:anyURI	This is the MSISDN of the mobile device to locate. Repeat the address parameter for multiple devices. The protocol and '+' identifier must be used for MSISDN, and must be URL-escaped. %3A represents ':' %2B represents '+'	No
requestedAccuracy	xsd:int	This is the preferred accuracy of the result, in metres. Typically, when you request an accurate location it will take longer to retrieve than a coarse location. For example, requestedAccuracy=10 will take longer than requestedAccuracy=100. ! Whilst it is mandatory to enter a value for this parameter, it will be ignored by U.S.Cellular Location service.	No

3.2.3 Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 1234
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"terminalLocationList":
        {"terminalLocation": [
                { "address": "tel:+16309700001",
                "currentLocation": {
```



```
                        "accuracy": "100",
                        "altitude": "0",
                        "latitude": "-80.86302",
                        "longitude": "41.277306",
                        "timestamp": "2009-06-04T02:51:39.000Z"},
                "locationRetrievalStatus": "Retrieved"}
]}}
```


3.2.4 Response Parameters

Table 2: Query Location Response Parameters (terminalLocationList type)

Parameter	Data Type	Description	Optional
terminalLocation	TerminalLocation [1..unbounded]	Collection of terminal locations (See table 3 below for details.)	No

Table 3: Query Location Response Parameters (terminalLocation type)

Parameter	Data Type	Description	Optional
address	xsd:anyURI	This is the address of the terminal(s), as per RFC 3966 in international format.	No
currentLocation	LocationInfo	This type provides details on the current location of the terminal using: • accuracy (metres) • altitude (metres) (Optional) • latitude (decimal degrees, ISO 6709) • longitude (decimal degrees, ISO 6709) • timestamp (xsd:dateTime format)	Yes
locationRetrievalStatus	common:	Indicates the outcome of the	No

	RetrievalStatus	query. Possible values are: • "Retrieved" (success) • "NotRetrieved" (unable to retrieve) - if status is NotRetrieved, the currentLocation object will be omitted from the response • "Error" (error retrieving location, due to a service or policy exception). See Response Codes & Exceptions below.	

4 Response Codes & Exceptions

4.1 Response Codes

HTTP response codes are used to indicate:

- **200** – Success!
- **400** – Bad request; check the error message for details
- **401** – Authentication failure, check your authentication details
- **403** – Forbidden; please provide authentication credentials
- **404** – Not found: mistake in the host or path of the service URI
- **405** – Method not supported: for example you mistakenly used a HTTP GET to create an SMS instead of a POST
- **500** – The server encountered an unexpected condition. This could be incorrect authentication details or limited user permission
- **503** – Server busy and service unavailable. Please retry the request.

For more details on these, refer to http://www.ietf.org/rfc/rfc2616.txt.

4.2 Exceptions

```
HTTP/1.1 400 Bad Request
```



```
Content-Type: application/json
Content-Length: 1234
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"requestError": {
 "serviceError": {
 "messageId": "SVC0002",
 "text": " Invalid input value for message part %1",
 "variables": " tel:+16309700000"
 }
}}
```

This section lists the available error codes, the possible reasons why the exception may have occurred, and possible solutions.

4.2.1 Service Exceptions

The following exceptions may be thrown when an operation fails:

Table 4: Service Error Codes

Error	Explanation
SVC0001 – Service error occurred	A service-related error has occurred as a result of a client invocation on the service. This category can be used for implementation-specific errors. Contact the support team.
SVC0002 – Invalid input value	An input parameter value is not of the expected type. Check the parameter types and re-submit your request.
SVC0004 – No valid address(es)	The requested terminal device address does not exist. Use an address that exists.

4.2.2 Policy Exceptions

A policy exception means that the request syntax is valid, however an operator policy has been broken.

The two types of policy exceptions are as follows:

- **POL-002**: Privacy Error **-** There was a problem using the Privacy service. Check your method use and re-submit your request
- **POL0001**: Policy error occurred. This exception may be thrown to indicate a fault relating to a policy associated with the service. This category can be used for implementation-specific errors such as:

Table 5: Policy Error Codes

Error	Explanation
POL-006: TPA exceeded its maximum allowed rate of transactions	The maximum rate of transactions is exceeded. Ensure that the rate of your requests is within the limits set up in your SLA, e.g. 10 TPS (Transactions Per Second).
POL-008: TPA is invalid	The Third Party Application authentication details are incorrect. Check your basic authentication username and password are correct and re-submit your request.
POL-014: White List is enforced, and address is not in White List	A white list is enforced and the number is not in the white list. Check your SLA details.
POL-016: Max Requests is enforced, and max requests has been exceeded	The maximum number of requests for this service is exceeded. Contact the support team.
POL-017: Operation is not allowed	The method/operation is not supported in your current SLA. Check your SLA and use a method that is supported.
POL-018: All targets were rejected for MDN access and authorization failure	This indicates that none of the destination numbers can be retrieved by the internal address resolver such as LDAP or Lookup. It includes white/black list rejection when the destination number cannot be found in either list that is enforced. In this case, check your policy contract and request the number to be added to/removed from the appropriate list.
POL-028: Black List is enforced, and address is in	A black list is enforced and the number is in the black

Error	Explanation
Black List	list. Check you SLA details.
POL-040: Max Destination Addresses is enforced and maximum destination addresses has been exceeded	A maximum destination address limit is enforced and it has been exceeded. Check your SLA for the limit and re-submit your request.
POL-042: The requested accuracy is less than the allowable value	The accuracy requested is too high. Re-submit the request with a lower accuracy, i.e. a value of 5000m or more.
POL-049: SPID Black List is enforced and address SPID is in the SPID Black List.	Applicable in multiple carrier deployments, Black List is enforced and the carrier identified by the Service Provider ID is in the black list. Therefore all the addresses from the carrier are rejected.

A Location Sandbox Service

The sandbox service replicates real U.S. Cellular Open API Location web service and returns response objects, or 'canned responses', against pre-configured subscriber CTN values. It does not connect to any external interface. Developers can use this service to test different scenarios of their application without connecting to the real subscriber profile service.

Developers should use the SandboxDataService API to preconfigure responses for subscriber addresses. Details are described in a separate SandboxDataService API guide.

Code examples are provided below.

Terminal Location Sandbox Request Example

```
GET
https://developerportal.uscellular.com/services/getLocationSandbox/2_0/location/queries/location?address=tel%3A%2B12345600001&requestedAccuracy=5000 HTTP/1.1

Authorization: Basic QWVwb25hVGdfghfdghUWF6eHN3MjNl

Host: developerportal.uscellular.com

Connection: Keep-Alive

User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

