



API MONETIZATION PLATFORM 4.0/4.1

AGILE SERVICE ENABLEMENT 1.0

OneAPI v2.0 Terminal Location REST Document Version 1.1

Document Properties

Document ID	OneAPI v2.0 Terminal Location REST
Document Version	1.1
Approval Date	22 March 2013
Originator	TWS team
Approver	Jim Beggs
Version Information	Updated from v1.0 issued on 19 Nov 2012 with AMp 4.0/4.1

Copyright

2015

© Aepona Limited,

Beacon House,

Clarendon Dock,

Belfast BT1 3BG

All rights reserved. This document or any part thereof may not, without the written consent of Aepona Limited, be copied, reprinted or reproduced in any material form including but not limited to photocopying, transcribing, transmitting or storing it in any medium or translating it into any language, in any form or by any means, be it electronic, mechanical, xerographic, optical, magnetic or otherwise.

The information contained in this document is proprietary and confidential and all copyright, trademarks, trade names, patents and other intellectual property rights in the documentation are the exclusive property of Aepona Limited unless otherwise specified. The information (including but not limited to data, drawings, specification, documentation, software listings, source or object code) shall not at any time be disclosed directly or indirectly to any third party without Aepona Limited's prior written consent.

The information contained herein is believed to be accurate and reliable. Aepona Limited accepts no responsibility for its use by any means or in any way whatsoever. Aepona Limited shall not be liable for any expenses, costs by damage that may result from the use of the information contained within this document. The information contained herein is subject to change without notice.

Table of Contents

1	Terminal Location REST Overview	5
2	Authentication.....	5
3	Methods	5
3.1	URIs	5
3.2	Query the Location of One Terminal.....	6
3.2.1	Request.....	6
3.2.2	Request Parameters.....	6
3.2.3	Response	7
3.2.4	Response Parameters	7
3.3	Query the location of Multiple Terminals	8
3.3.1	Request.....	8
3.3.2	Request Parameters.....	8
3.3.3	Response	9
3.3.4	Response Parameters	10
4	Response Codes & Exceptions	11
4.1	Response Codes	11
4.2	Exceptions.....	11
4.2.1	Service Exceptions	11
4.2.2	Policy Exceptions	12

1 Terminal Location REST Overview

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

! Throughout this document, the examples may be shown WITHOUT URL encoding for readability purposes, e.g. if the address "tel:+123456789" is in the URL example, this should be encoded as "tel%3A%2B123456789", 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 – section 3.2
- Query the location of Multiple Terminals – section 3.3

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
https://{serverRoot}/{apiVersion}/location/queries/location?address={address}&requestedAccuracy={metres}
- Query Multiple Location URI
https://{serverRoot}/{apiVersion}/location/queries/location?address={address1}&address={address2}&address={address3}&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
serverRoot	Server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: <i>http://example.com</i> , <i>http://developer.aepona.com/TerminalLocationService/OneAPI_REST_v2.0/sandbox</i>
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://example.com/2_0/location/queries/location?address=tel%3A%2B1630
9700001&requestedAccuracy=5000 HTTP/1.1
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	No

		accurate location it will take longer to retrieve than a coarse location. For example, requestedAccuracy=10 will take longer than requestedAccuracy=100.	
--	--	--	--

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": "1001.0",
        "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 2.1 below for details).	No

Table 2.1: 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	Yes

		location of the terminal using: <ul style="list-style-type: none"> • accuracy (metres) • altitude (metres) (Optional) • latitude (decimal degrees, ISO 6709) • longitude (decimal degrees, ISO 6709) • timestamp (xsd:dateTime format). 	
locationRetrievalStatus	common:RetrievalStatus	Indicates the outcome of the query. Possible values are: <ul style="list-style-type: none"> • "Retrieved" (success) • "NotRetrieved" (unable to retrieve) - if status is NotRetrieved, the current-Location object will be omitted from the response • "Error" (error retrieving location, due to a service or policy exception). See Response Codes & Exceptions below. 	No

3.3 Query the location of Multiple Terminals

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

3.3.1 Request

```
GET
https://example.com/2_0/location/queries/location?address=tel%3A%2B1630
9700001&address=tel%3A%2B16309700002&requestedAccuracy=5000
Accept: application/json
```

3.3.2 Request Parameters

Table 3: 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. In the sample code above, the location of 2	No

		<p>terminals has been requested.</p> <p>The protocol and '+' identifier must be used for MSISDN, and must be URL-escaped.</p> <p>%3A represents ':'</p> <p>%2B represents '+'</p>	
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.	No

3.3.3 Response

```
{
  "terminalLocationList": {
    "terminalLocation": [
      {
        "address": "tel:+16309700001",
        "currentLocation": {
          "accuracy": "100",
          "altitude": "1001.0",
          "latitude": "-80.86302",
          "longitude": "41.277306",
          "timestamp": "2009-06-04T02:51:35.000Z"
        },
        "locationRetrievalStatus": "Retrieved"
      },
      {
        "address": "tel:+16309700002",
        "errorInformation": {
          "serviceException": {
            "messageId": "SVC0001",
            "text": "A service error occurred. %1 %2",
            "variables": [
              "Location information is not available for",
              "tel:+16309700002"
            ]
          }
        },
        "locationRetrievalStatus": "Error"
      }
    ]
  }
}
```

```
}
}
```

3.3.4 Response Parameters

Table 4: Query Location - Response Parameters (terminalLocationList type)

Parameter	Data Type	Description	Optional
terminalLocation	TerminalLocation [1..unbounded]	Collection of terminal locations. (See table 5 for details).	No

Table 5: 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: <ul style="list-style-type: none"> accuracy (metres) altitude (metres) (Optional) latitude (decimal degrees, ISO 6709) longitude (decimal degrees, ISO 6709) timestamp (xsd:dateTime format). 	Yes
locationRetrievalStatus	common:RetrievalStatus	Indicates the outcome of the query. Possible values are: <ul style="list-style-type: none"> "Retrieved" (success) "NotRetrieved" (unable to retrieve) "Error" (error retrieving location, due to a service or policy exception). 	No
errorInformation	common:ServiceError	Indicates the reason for the error if locationRetrievalStatus=Error. See Response Codes & Exceptions below.	Yes

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:+016309700000"
  }
}}
```

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 5: Service Exceptions

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:

- **POL0002:** 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 6: 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-015: Black List is enforced, and address is in Black List	A black list is enforced and the number is in the black list. Check you SLA details.

Error	Explanation
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	<p>This indicates that none of the destination numbers can be retrieved by the internal address resolver such as LDAP or Lookup.</p> <p>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.</p>
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.