API MONETIZATION PLATFORM VODAFONE INDIA 1.0

Agile Services Enablement 2.0

Sandbox API Guide

Document Version 3.0

This Sandbox API Guide has been prepared for Vodafone India.

Copyright © Intel Corporation 2015; all rights reserved.

Document Revision History

Rev #	Date	Description
1.0	1 November 2013	Initial issue of Sandbox Guide for API services (Payment, SMS and Terminal Location)
2.0	19 February 2014	Updated with the addition of Device Details, DoNotDisturb sandbox services, and support for Payment refund operation.
2.1	4 April 2014	Updated with the addition of SIMX sandbox service.
3.0	15 September 2014	Updated with the addition of Balance Check sandbox service.

Table of Contents

1 Overview 5
1.1 References 5
2 Authentication 5
3 Sandbox URI 5
4 Balance Check Sandbox 7
4.1 Canned Responses for Predefined Parameter Values 7
5 Device Details Sandbox 8
5.1 Canned Responses for Predefined Parameter Values 8
6 DoNotDisturb Sandbox Service 9
6.1 Canned Responses for Predefined Parameter Values 9
7 SIMX Sandbox Service 10
7.1 Canned Responses for Predefined Parameter Values 10
7.1.1 SIMX – Policy Exception Response – POL0001 11
7.1.2 SIMX – Service Exception Response – SVC0002 11
8 Payment Sandbox 12
8.1 Canned Responses for Predefined Parameter Values 12
8.2 Canned Response Error Examples - Payment 14
8.2.1 Charge – TOO_SHORT Response – Service Error SVC0002 14
8.2.2 Charge – Invalid Code Response – Service Error SVC0002 14
8.2.3 Charge – Policy Exception Response – POL0001 15
8.2.4 Get Transaction Status Request – Invalid Transaction ID 15
8.2.5 Get Transaction Status Response – Invalid Transaction ID 15
9 SMS Sandbox 16
9.1 Canned Responses for Predefined Addresses 16
9.2 Canned Response Error Examples - SMS 17
9.2.1 Send SMS – Service Error Response 17
9.2.2 Send SMS – Policy Error Response 18
9.3 Canned Response Examples - getSmsDeliveryStatus 18
9.3.1 Get SMS Delivery Status Request 19
9.3.2 Get SMS Delivery Status Response 19
9.4 DeliveredToTerminal Delivery Receipt Notifications 19
9.4.1 Delivery Notification Format (Request) 19

9.4.2 Delivery Notification Response....20
10 Terminal Location Sandbox 21
10.1 Canned Responses for Predefined Addresses 21
10.2 Terminal Location Service Sandbox Properties 22
10.3 Canned Response Error Examples – Terminal Location....23
10.3.1 Get Location – Service Error Response 23
10.3.2 Get Location – Policy Error Response....23



1 Overview

This document describes the Sandbox service provided for AMP-VFI v1.0 ASE APIs, which comprises sets of 'canned responses' which simulate key operations for testing purposes. Calls are made to the Sandbox server, and bypass the production server. This service is available for the following RESTful APIs and selected methods:

- *AMP-VFI Balance Check* – described in section 4
- *AMP-VFI Device Details* – described in section 5
- *AMP-VFI Do Not Disturb* – described in section 6
- *AMP-VFI SIMX* – described in section 7
- *OneAPI 2.1 Payment* – charge and getTransaction methods - described in section 8
- *OneAPI 2.0 SMS* – sendSMS and getSmsDeliveryStatus methods -described in section 9
- *OneAPI 2.0 Terminal Location* – get single or multiple locations - described in section 10

Each section contains a list of 'canned responses' and the parameter values to use in the requests to elicit those fixed responses, with any other information necessary to utilise the Sandbox service. Where applicable, sandbox service specific properties and exceptions are also provided.

1.1 References

Service URIs, general example codes, error responses on validation and policy enforcement that apply to the API services are referenced out to the following API Guides:

- *Voda-BalanceCheck_API*
- *Voda-DeviceDetails_API*
- *Voda-DoNotDisturb_API*
- *Voda-Payment_API*
- *Voda-SIMX_API*
- *Voda-SMS_API*
- *Voda-TerminalLocation_API*

Common information on API and usage are captured in:

- *Voda-OneAPI-Common-Information-Guide*

2 Authentication

HTTP Basic Authentication is required. If HTTPS is used, a server side is also required.

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

3 Sandbox URI

The URI to use in order to access the Sandbox service can be obtained from the Vodafone Partner Portal.

Log in and navigate to *Dashboard > Manage Partner > Manage Project > Testing Environment*

The list of available services and endpoints for your project will be displayed on the page. These endpoints are valid only whilst your project is in Sandbox status.

4 Balance Check Sandbox

This section describes the Sandbox feature for Balance Check service.

4.1 Canned Responses for Predefined Parameter Values

The table below lists the default addresses configured in the Sandbox to return the expected results. Address validation is enforced in such a way that if any address in the request is invalid, the entire request is rejected and the applicable error response is returned.

? Using any other valid destination address not pre-defined for an expected result should return a canned successful response. The responses are fixed, and will not reflect the actual values input.

? If TPS policy is applied to the Sandbox, the policy will be validated against any MSISDN used for the sandbox call.

Table 1: Balance Check Sandbox – Pre-defined Addresses and Results

Parameter	Value	Expected Result
msisdn	any correctly formatted number	Successful response. The default balance and range configurations are set up for the sandbox to return the balance range value 4 for the balance of Rs.45.
msisdn	tel:+917838157190	SVC0004 service error response: Invalid address with reason, Subscriber Profile not found.
msisdn	tel:+917838157193	POL-008 policy error response: TPA is Invalid.
msisdn	tel:+917838157195	POL-006 policy error response: TPA exceeded its maximum allowed rate of transactions.
msisdn	tel:+917838157197	SVC0001 service error response: Out of range error occurred.
msisdn	tel:+917838157199	SVC0001 General service error.

5 Device Details Sandbox

This section describes the Sandbox feature for Device Details service.

5.1 Canned Responses for Predefined Parameter Values

The table below lists the default addresses configured in the Sandbox to return the expected results. Address validation is enforced in such a way that if any address in the request is invalid, the entire request is rejected and the applicable error response is returned.

? Using any other valid destination address not pre-defined for an expected result should return a canned successful response. The responses are fixed, and will not reflect the actual values input.

Table 2: Device Details Sandbox – Pre-defined Addresses and Results

Parameter	Value	Expected Result
endUserId	+910000001000 +910000001006	Successful response
endUserId	+910000001001	Response with error code: ADDRESS_BLACK_LIST_VIOLATION
endUserId	+910000001002	Response with error code: ADDRESS_WHITE_LIST_VIOLATION
endUserId	+910000001003	Response with error code: ADDRESS_AUTHORISATION_VIOLATION
endUserId	+910000001004	Response with error code: MAX_NUMBER_TRANSACTIONS_VIOLATION
endUserId	+910000001005	Response with error code: MAX_NUMBER_TRANSACTIONS_PER_INTERVAL_VIOLATION

6 DoNotDisturb Sandbox Service

This section describes the Sandbox feature for DoNotDisturb service.

6.1 Canned Responses for Predefined Parameter Values

The table below lists the default addresses configured in the Sandbox to return the expected results. Address validation is enforced in such a way that if any address in the request is invalid, the entire request is rejected and the applicable error response is returned.

? Using any other valid destination address not predefined for an expected result should return a canned successful response. The responses are fixed, and will not reflect the actual values input.

Table 11: DoNotDisturb Sandbox – Pre-defined Addresses and Results

Parameters	Values	Expected Result
endUserId	+910000001006	Successful response to Get all DND category status request
endUserId	+910000001001	Response with error code: ADDRESS_BLACK_LIST_VIOLATION
endUserId	+910000001002	Response with error code: ADDRESS_WHITE_LIST_VIOLATION
endUserId	+910000001003	Response with error code: ADDRESS_AUTHORISATION_VIOLATION
endUserId	+910000001004	Response with error code: MAX_NUMBER_TRANSACTIONS_VIOLATION
endUserId	+910000001005	Response with error code: MAX_NUMBER_TRANSACTIONS_PER_INTERVAL_VIOLATION

7 SIMX Sandbox Service

This section describes the Sandbox feature for SIMX service.

7.1 Canned Responses for Predefined Parameter Values

The table below lists the default addresses and days values configured in the Sandbox to return the expected results. Address validation is enforced in such a way that if any address in the request is invalid, the entire request is rejected and the applicable error response is returned.

? The responses are fixed, and do not necessarily reflect the actual values input.

Table 11: SIMX Sandbox – Pre-defined Parameter and Results

Parameters	Values	Expected Result
endUserId days	tel:+919999268188 1 or 2	Successful response, with value 0 indicating SIM exchange had not occurred
endUserId days	tel:+919999268188 4, 5, 6, or 7	Successful response, with value 1 indicating SIM exchange occurred
endUserId days	tel:+919999268220 1, 2, 3, 4, 5, 6 or 7	Successful response, with value 2 indicating unknown subscriber
endUserId days	tel:+919999268188 8 or greater	Policy error response, for violation of max days range: POL-019: Input Value Exceeds Maximum Allowed
endUserId days	tel:+919999268188 0 or less, including -1. etc	Policy error response, for violation of min days range: POL-020: Input Value Exceeds Minimum Allowed
endUserId days	tel:+919825020006 1	POL-006 policy error response: TPA exceeded its maximum allowed rate of transactions
endUserId days	tel:+919825942918 1	POL-008 policy error response for authentication error: TPA is invalid
endUserId	Any invalid address that cannot be normalized	Service error response, SVC0004 with text: No valid address(es). Invalid examples include addresses -

		• containing letters (upper case O in place of 0) • too long (eg, 13 digits), too short (eg, 8 digits prefixed with tel:+) For further information on address normalization, see the Common Information Guide.
endUserId days	tel:+919999268188 Invalid days, eg 5.5, not present, abc, etc	Service error response, SVC0002, with text: Invalid input value and additional text describing the issue.

7.1.1 SIMX – Policy Exception Response – POL0001

The example below shows the format of a policy error against the preconfigured endUserAddress tel:+919999268188 and the days value 8, which will return a max value exceeded response.

```
{
   "requestError": {
      "policyException": {
         "messageId": "POL0001",
         "text": "A policy error occurred. Error code is POL-019: Input Value Exceeds Maximum Allowed.",
         "variables": [
            "POL-019",
            "Input Value Exceeds Maximum Allowed."
         ]
      }
}
```

7.1.2 SIMX – Service Exception Response – SVC0002

The example below shows the format of a service error against the preconfigured endUserAddress tel:+919999268188 and an absent days value.

```
{
"requestError": {
"serviceException": {
"messageId": "SVC0002",
"text": " Invalid input value. 'days' parameter must be specified."
}
}
}
```

8 Payment Sandbox

This section describes the Sandbox feature for Payment service, in the following sub-sections:

- Canned Responses for Pre-defined parameter values in Charge/getTransaction call - below
- Canned Response Error Examples - section 4.2

8.1 Canned Responses for Predefined Parameter Values

The table below lists endUserId values configured in the Sandbox to return expected results. Values for the parameters in the body of the message are open, except where specified.

End User Id validation is enforced in such a way that if any address in the request is invalid, the entire request is rejected and the applicable error response is returned.

? Using any other valid address not pre-defined for an expected result should return a successful response. The responses are fixed, and will not reflect the actual values input.

The values in the table below apply to charge, refund and getTransaction methods, except for those marked with an asterisk*, which apply to getTransaction method only.

Table 3: Charge/Refund/getTransaction – Pre-defined Parameters and Results

Parameters	Values	Expected Result
endUserId *and* transactionOperationStatus	tel:+919825010000 or any other valid address not pre-defined. Charged	Success – Charged
endUserId *and* transactionOperationStatus	tel:+919825010000 or any other valid address not pre-defined. Refunded	A refund transaction will be created and a response with transactionOperationStatus = Pending will be returned; after one minute, a notification is returned, in the same format as the response, with transactionOperationStatus = Refunded.
endUserId	tel:+ 919825010500	REFUND_PROCESSING *Note: This scenario applies to getTransaction call only.
endUserId	tel:+ 919825010501	REFUND_REFUNDED

		*Note: This scenario applies to getTransaction call only.
endUserId	tel:+ 919825010502	REFUND_REFUSED *Note: This scenario applies to getTransaction call only.
endUserId	tel:+12345678	Unsuccessful - IS_POSSIBLE
endUserId	tel:+1234567	Unsuccessful - TOO_SHORT
endUserId	tel:+123456789	Unsuccessful - TOO_LONG
endUserId	tel:+919825010001	SVC0001: Service error occurred
endUserId	tel:+919825010002	SVC0002: Invalid input value
endUserId *and* transactionId	Tel:+919825010000 or any other valid address not pre-defined NONEXISTENT	SVC0002: Invalid input value *Note: This scenario applies to getTransaction call only.
endUserId *and* code	Tel:+919825010000 or any other valid address not pre-defined INVALID	SVC0002: Invalid input value
endUserId	tel:+919825010003	SVC0004: No valid address(es)
endUserId	tel:+919825020006	POL-006: TPA exceeded its maximum allowed rate of transactions
endUserId	tel:+919825020008	POL-008: TPA is invalid
endUserId	tel:+919825020010	POL-010: Subscriber target not authorized
endUserId	tel:+919825020014	POL-014: Destination White List is enforced, and address is not in Destination White List
endUserId	tel:+919825020016	POL-016: Max Requests is enforced, and max requests has been exceeded

endUserId	tel:+919825020017	POL-017: Operation is not allowed
endUserId	tel:+919825020028	POL-028: Destination Black List is enforced, and address is in Destination Black List
endUserId	tel:+919825020040	POL-040: Max Destination Addresses is enforced and maximum destination addresses has been exceeded

8.2 Canned Response Error Examples - Payment

This section lists a few of the canned responses returned for charge and getTransaction requests.

8.2.1 Charge – TOO_SHORT Response – Service Error SVC0002

The example below shows the format of a service error, in this case, TOO_SHORT, against the pre-defined end user Id tel:+1234567.

```
{
requestError: {
serviceException: {
messageId: "SVC0002"
text: "Invalid input value for message part address with value tel:+1234567. Reason TOO_SHORT"
variables: [3]
0: "address"
1: "tel:+1234567"
2: "TOO_SHORT"
}
}
}
```

8.2.2 Charge – Invalid Code Response – Service Error SVC0002

The example below shows the format of a service error against the code/event Id set as INVALID, with a valid end user Id for a successful response.

```
{
requestError: {
serviceException: {
messageId: "SVC0002"
text: "Invalid input value [INVALID] for parameter [code]"
}
}
}
```

8.2.3 Charge – Policy Exception Response – POL0001

The example below shows the format of a policy error against the preconfigured endUserAddress tel:+919825020008 which will return a TPA invalid response.

```
{
requestError: {
policyException: {
messageId: "POL0001"
text: "A policy error occurred. Error code is POL-008: TPA is invalid",
variables : ["POL-008", "TPA is invalid"]
}
}
}
```

8.2.4 Get Transaction Status Request – Invalid Transaction ID

To test an invalid transaction Id scenario, edit the resource URL with a valid end user Id for a successful response and transactionId as defined in the table above, as shown below:

```
GET http://example.com/2_1/payment/tel:+919825010000/transactions/amount/NONEXISTENT
HTTP/1.1
Authorization: Basic<base64 encoded application credentials>
```

8.2.5 Get Transaction Status Response – Invalid Transaction ID

```
{
requestError: {
serviceException: {
messageId: "SVC0002"
text: "Invalid input value [NONEXISTENT] for parameter [transactionId]"
}
}
}
```

9 SMS Sandbox

This section describes the Sandbox feature for SMS service, in the following sub-sections:

- Canned Responses for Pre-defined parameter values in SendSMS/getSmsDeliveryStatus call - below
- Send SMS Example Error Responses - section 7.2
- getSmsDeliveryStatus call examples – section 7.3
- DeliveredToTerminal notification format – section 7.4

9.1 Canned Responses for Predefined Addresses

The table below lists the default addresses configured in the Sandbox to return the expected results. Address validation is enforced in such a way that if any address in the request is invalid, the entire request is rejected and the applicable error response is returned.

? Using any other valid destination address not pre-defined for an expected result should result in a Delivery Impossible status being returned, for sendSMS or getSmsDeliveryStatus call.

? The responses are fixed, and will not reflect the actual values input.

- If *modify sender address* policy is enabled, the sender address in the request will be defaulted or overridden to the default sender address, which is +912233331017.
- If *modify sender name* policy is enabled, the sender name, in the request will be defaulted or overridden to the default sender name, which is the Canned Response Sender.

The addresses specified for the first three results (marked with an asterisk*) can be used to test getSmsDeliveryStatus, with the expected result as defined. See section 7.3 for details and examples.

Using the address marked with # will also trigger a notification to be sent to the application at the notifyURL. See section 7.4 for details and examples.

Table 4: Send SMS – Pre-defined Addresses and Results

Type of Address	Address	Expected Result
DestinationAddress	tel:+912233331000	Success - DeliveredToNetwork*
DestinationAddress	tel:+912233331018	Success - DeliveredToTerminal*
DestinationAddress	Tel:+912233331001 or any other valid address not pre-defined	Success - DeliveryImpossible*#
DestinationAddress	tel:+07766544732	Unsuccessful - INVALID_COUNTRY_CODE

DestinationAddress	tel:+12345678	Unsuccessful - IS_POSSIBLE
DestinationAddress	tel:+1234567	Unsuccessful - TOO_SHORT
DestinationAddress	tel:+123456789	Unsuccessful - TOO_LONG
DestinationAddress	tel:+912233331002	Sender Blacklist Exception
DestinationAddress	tel:+912233331003	Sender Whitelist Exception
DestinationAddress	tel:+912233331004	Maximum Transactions Per Interval exception
DestinationAddress	Tel:+912233331005	Maximum Transactions Allowed exception
DestinationAddress	tel:+912233331006	Operation Set - sendSms exception
DestinationAddress	tel:+912233331007	Modify Sender Address
DestinationAddress	tel:+912233331008	Modify Sender Name
DestinationAddress	tel:+912233331009	Destination Address Blacklist exception
DestinationAddress	tel:+912233331010	Destination Address Whitelist exception
DestinationAddress	tel:+912233331011	HTTPS Callback Required exception
DestinationAddress	tel:+912233331012	Maximum Destination Address exception
DestinationAddress	tel:+912233331013	Maximum Message Length exception
DestinationAddress	tel:+912233331014	Minimum Message Length exception
DestinationAddress	tel:+912233331015	Receipting exception
DestinationAddress	tel:+912233331016	Authentication exception

9.2 Canned Response Error Examples - SMS

This section lists a few of the canned responses returned for sendSMS requests.

9.2.1 Send SMS – Service Error Response

The example below shows the format of a service error response, in this case, INVALID_COUNTRY_CODE against the destination address.

```
{
  "requestError" : {
```

```
    "serviceException" : {
      "messageId" : "SVC0002",
      "text" : "Invalid input value for message part address with value tel:+07766544732. Reason INVALID_COUNTRY_CODE",
      "variables" : [ "address", "tel:+07766544732", "INVALID_COUNTRY_CODE" ]
    }
  }
}
```


9.2.2 Send SMS – Policy Error Response

The example below shows the format of a policy error response, in this case, POL-021, minimum message length violation.


```
Response-Code: 403
Content-Type: application/json
Headers: {Content-Type=[application/json], Date=[Wed, 21 Aug 2013 09:07:09 GMT]}
{
 "requestError" : {
    "policyException" : {
      "messageId" : "POL0001",
      "text" : "A policy error occurred. Error code is POL-021: Min Message Length is enforced and message length is less than minimum
value.",
      "variables" : [ "POL-021", "Min Message Length is enforced and message length is less than minimum value.", null ]
      }
  }
}
```


9.3 Canned Response Examples - getSmsDeliveryStatus

Sandbox is available to test a getSmsDeliveryStatus call, to follow a sendSms call. You can send SMS to single addresses, or to multiple addresses, and get their delivery statuses by using the resource URL returned in the successful response.

Sandbox is available to test a getSmsDeliveryStatus call, to follow your sendSms call. You can send SMS to single addresses, or to multiple addresses, and get their delivery statuses by using the resource URL returned in the successful response.

As noted above the table of addresses/results, the addresses configured for the following results are valid for this sandbox method:

- DeliveredToNetwork
- DeliveredToTerminal
- DeliveryImpossible

Other valid addresses, which are not configured for any other canned responses, can also be used to return a DeliveryImpossible status confirmation.

Use GET command and the URI: [resource URL returned in the Successful response to your send SMS]/deliveryInfos

9.3.1 Get SMS Delivery Status Request

An example using the resource URL from a successful response as in section 7.1 above:

```
HTTP Method : GET
Address: http://example.com/sendSms/2_0/smsmessaging/outbound/tel:+912233331017/requests/4e8b69ba-2b21-4a5b-b2ac-bdf354f2ea2d/deliveryInfos
Encoding: UTF-8
Content-Type: application/json
Accept: application/json
Headers: {Accept=[application/json], Authorization=[Basic VGVzdDpUZXN0]}
```

9.3.2 Get SMS Delivery Status Response

```
Response-Code: 200
Content-Type: application/json
{
"deliveryInfoList" : {
"resourceURL" : "http://example.com/smsSandbox/2_0/smsmessaging/outbound/tel:+912233331017/requests/4e8b69ba-2b21-4a5b-b2ac-bdf354f2ea2d/deliveryInfos",
"deliveryInfo" : [ {
"address" : "tel:+912233331000",
"deliveryStatus" : "DeliveredToNetwork"
} ]
}
}
```

9.4 DeliveredToTerminal Delivery Receipt Notifications

When a sandbox sendSms request is made to an address configured to return *Success – DeliveredToTerminal* response, in other words, to tel:+912233331018 in the default configuration, a delivery receipt will be sent to your application at the notifyURL provided in the request.

9.4.1 Delivery Notification Format (Request)

```
{
  "deliveryInfoNotification" : {
    "deliveryInfo" : [ {
      "address" : "tel:+912233331018",
      "deliveryStatus" : "DeliveredToTerminal"
    } ]
  }
```

```
}
```

9.4.2 Delivery Notification Response

```
HTTP/1.1 204 No Content
```

10 Terminal Location Sandbox

This section describes the Sandbox feature for Terminal Location service, in the following sub-sections:

- Canned Responses for Predefined Addresses - below
- Sandbox Service Default Properties – section 6.2
- Get Location Example Error Responses - section 6.3

10.1 Canned Responses for Predefined Addresses

The table below lists the default addresses configured in the Sandbox to return the expected results. Address validation is enforced in such a way that if any address in the request is invalid, the entire request is rejected and the applicable error response is returned.

? Using any other valid destination address not pre-defined for an expected result should return a canned successful response. The responses are fixed, and will not reflect the actual values input.

Table 5: TL Sandbox - Pre-defined Addresses and Results

Type of Address	Address	Expected Result
DestinationAddress	Tel:+912233331000 or any other valid address not pre-defined	Success - Retrieved
DestinationAddress	Tel:+912233331001 or any other valid address not pre-defined	Success - Retrieved
DestinationAddress	tel:+912233331002	Unsuccessful - NotRetrieved
DestinationAddress	07766544732	Unsuccessful - INVALID_COUNTRY_CODE
DestinationAddress	tel:+12345678	Unsuccessful - IS_POSSIBLE
DestinationAddress	tel:+1234567	Unsuccessful - TOO_SHORT
DestinationAddress	tel:+123456789	Unsuccessful - TOO_LONG
DestinationAddress	tel:+912233331003	Unsuccessful - Blacklist Destination Address
DestinationAddress	tel:+912233331004	Unsuccessful - Whitelist Destination Address Policy
DestinationAddress	tel:+912233331005	Unsuccessful - Maximum Destination Address Policy

DestinationAddress	tel:+912233331006	Unsuccessful - Maximum Transaction per Interval
DestinationAddress	tel:+912233331007	Unsuccessful - Maximum Number of Transaction
DestinationAddress	tel:+912233331008	Unsuccessful - Operation Set Policy
DestinationAddress	tel:+912233331009	Unsuccessful - Authentication Exception

10.2 Terminal Location Service Sandbox Properties

The table below shows the default service property values configured for the Terminal Location Sandbox Service.

Table 6: Terminal Location Service Sandbox Properties

Property Name	Data Type	Optional	Default Value	Description
pmeServiceId	string	No		The PME Service ID
publicAddress	string	No		The public address appended to the endpoint URL of the service e.g. tlx, as configured on the Portal.
sandboxMode	string	Yes	CannedResponse	The sandbox mode. Possible values are CannedResponse and SandboxProvisioningService.
tolerance	string	Yes	DelayTolerant	The tolerance service property. Qualified values are DelayTolerant, NoDelay and LowDelay. The default value, DelayTolerant, is used when no value is set.
acceptableAccuracy	int	Yes	100000	The acceptable accuracy service property. Location Info is not retrieved for a request with a requestedAccuracy value greater than the defined acceptable accuracy. A Service Error is thrown when acceptableAccuracy value is less than the actual accuracy in the response for a specific address. See Service Exceptions below.
minimumAccuracy	int	Yes	1	The minimum accuracy to be requested. minimum accuracy must be <= to requestedAccuracy otherwise, an

				exception is thrown. See Service Exceptions below.

10.3 Canned Response Error Examples – Terminal Location

This section lists a few of the canned responses returned for getLocation requests.

10.3.1 Get Location – Service Error Response

The example below shows the format of a service error response, in this case, the address in the request was too short.

```
{
  "requestError" : {
    "serviceException" : {
      "messageId" : "SVC0002",
      "text" : "Invalid input value for message part address with value tel:+1234567. Reason TOO_SHORT",
      "variables" : [ "address", "tel:+1234567", "TOO_SHORT" ]
    }
  }
}
```

10.3.2 Get Location – Policy Error Response

The example below shows the format of a policy error response, in this case, an authentication failure.

```
Response-Code: 401
Content-Type: application/json
Headers: {Content-Type=[application/json], WWW-Authenticate=[Basic realm=], Date=[Thu, 22 Aug 2013 12:31:09 GMT]}
{
  "requestError" : {
    "policyException" : {
      "messageId" : "POL0001",
      "text" : "A policy error occurred. Error code is POL_TPA_INVALID.",
      "variables" : [ "TPA is invalid." ]
    }
  }
}
```

End of Document