U.S. Cellular

Open API Project

OneAPI v2.0 Multimedia Messaging REST

Document Version 1.1

Document Revision History

Rev #	Date	Description
1.0	June 24 2013	Initial version, based on Aepona OneAPI 2.0 MMS REST guide doc v1.2a.
1.1	Nov 1 2013	Updated URIs and examples to match production values. Added information sendMms Sandbox services.

Copyright

Table of Contents

1 MMS REST Overview

The MMS interface allows an application to send and receive MMS messages. You can find some examples of why you may want to do this in the use cases at http://www.oneapi.gsmworld.com.

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

! Data types indicated as XSD in parameter tables refer to the standard XSD type. Relevant information may be obtained from *http://www.w3.org/2001/XMLSchema*.

! 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

The following methods are available:

- Sending MMS:
    - Sending MMS from an Application
    - Query Delivery Status of an MMS
    - Notify Client About Outbound Message Delivery Status
- Receiving MMS:

- Retrieve List of Messages Sent to an Application (which is identified by **registrationId**)
- Retrieve URIs of Media Attached to a Received MMS
- Delete a Received MMS from Storage, instead of having to poll.

POST, GET and DELETE are used in OneAPI MMS.

? Representation formats – the MMS API supports multipart/form-data content type for sending an MMS and application/x-www-form-urlencoded and JSON content types for all other POST operations. The response content type is application/JSON.

3.1 Operations and URIs

The URIs for the resources are as follows:

- Sending MMS

  Send an MMS message from your Application to one or more mobile terminals – described in section 5.1

  POST https://developerportal.uscellular.com/services/sendMMS/{apiVersion}/messaging/outbound/{senderAddress}/requests

  Query the delivery status of the MMS identified by requestId – described in section 5.2

  GET https://developerportal.uscellular.com/services/sendMMS/{apiVersion}/messaging/outbound/{senderAddress}/requests/{requestId}/deliveryInfos

  A message delivery status notification is sent to the client by the server, with the format as described in section 5.3

- Sending MMS from your Application to the Sandbox mmsSend service – descrbed in Appendix D

  POST https://developerportal.uscellular.com/services/sendMMSSandbox/{apiVersion}/messaging/outbound/{senderAddress}/requests

- Receiving MMS

Retrieve MMS messages sent to your application (which is identified by **registrationId**) – described in section 6.1

GET https://developerportal.uscellular.com/services/getReceivedMms/{apiVersion}/messaging/inbound/registrations/{registrationId}/messages

Retrieve URIs pointing to the media file attachments on an MMS – described in section 6.2

GET https://developerportal.uscellular.com/services/getReceivedMms/{apiVersion}/messaging/inbound/registrations/{registrationId}/messages/{messageId}

Remove Message from storage – described in section 6.3

DELETE https://developerportal.uscellular.com/services/getReceivedMms/{apiVersion}/messaging/inbound/registrations/{registrationId}/messages/{messageId}

A message delivery status notification is sent to the client by the server, with the format as described in section 6.4.

? Message formats used in delivery and arrival status notifications are described in sections 5.3 and 6.4 as indicated above. These notifications are enabled from the Dev Portal, the operations for which are described in Appendix A.

The variables used in request URLs are described below:

Name	Description
{apiVersion}	Version of the API that the client wants to use. In this case 2_0.
{senderAddress}	Typically the SMS short code, which identifies the client application. If this is used as a parameter in the request body, the values must match.
{requestId}	The outbound message request ID generated by the server.
{registrationId}	The reference to the off-line retrieval criteria provisioned in advance and known to the application.

4 Encoding & Serialisation Details for MIME Format

A MIME multipart message used in the MMS API consists of several parts:

- The root structure (e.g. InboundMMSMesssage or OutboundMMSMessage), expressed in JSON or form url-encoded format for requests. This part conveys the origin, destination addresses, subject, priority, message identifier, etc.
- The multimedia contents or attachments expressed in the form of links or as MIME body parts, within the HTTP request or response. They include all contents, both plain text as well as other MIME types (images, videos, etc.), potentially exchangeable in MMSs.

For these messages with multiple body parts, multipart/form-data will be used instead, as per RFC2388 and HTML FORMS. This means that:

- Root fields as described above will be included as a single form field with a MIME body with:
  - Content-Disposition: form-data; name="root-fields"
  - Content-Type: <application/json or application/x-www-form-urlencoded>.
- Multimedia contents (text, images, etc.) will be included using one of the following options:
  - When the message contains only one content; by including a MIME body with:

    Content-Disposition: form-data; name="attachments", filename="<Name of the message content>"

    Content-Type:<Corresponding Content-Type>
  - When the message contains more than one content; by including a form-field with a MIME body with:

    Content-Disposition: form-data; name="attachments"

    Content-Type: multipart/mixed and then every one of the possible message contents included as subparts, using:

    Content-Disposition:attachment; filename="<Name of the message content>" Content-Type: <Corresponding Content-Type>
- For every HTTP body part and sub-parts, it is possible to include other parameters (Content-Description, Content-Transfer-Encoding), etc.

The following section contains examples of MIME multipart messages.

5 Sending MMS

5.1 Sending MMS from an Application

This section describes how to send MMS from an application.

5.1.1 Request

```
POST
https://developerportal.uscellular.com/services/sendMMS/2_0/messaging/outbound/12345/reque
sts HTTP/1.1

Content-Type: multipart/form-data;boundary="===============123456==";

Authorization: Basic QWVwb25hVfdghfdghdfghdfgWF6eHN3MjNl

Content-Length: 639

Host: developerportal.uscellular.com

Connection: Keep-Alive

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


--===============123456==

Content-Disposition: multipart/form-data; name=”root-fields”

Content-Type: application/json



{"outboundMessageRequest":

      {

    "address": ["tel:+14167001003"],

    "clientCorrelator": "45646",

    "receiptRequest":
{"notifyURL":"http://application.example.com/notifications/DeliveryInfoNotification "},

    "outboundMMSMessage": {"priority": "High", "subject": "FN-MMS-01"},

    "senderAddress": "12345",

      "senderName": "MyName"
```

```
     }
}
--================123456==
Content-Disposition: multipart/form-data; name=”attachments”
Content-Type: multipart/mixed; boundary=”===12345===”


--===12345===
Content-Disposition: attachments; filename=" dublin_bay.jpg" Content-Type: text/plain;
See attached photo
--===12345===
Content-Disposition: attachment; filename="dublin_bay.jpg"
Content-Type: image/jpeg
--===12345===--
--================123456==--
```

Use POST to create the MMS. The **senderAddress** in the URL must be URL-escaped.

5.1.2 Request Parameters

Table 1: Sending MMS from an Application - Request Parameters (outboundMessageRequest type)

Parameter	Type	Description	Optional
address	xsd:anyURI [1..unbounded]	There must be at least one **address** (CTN) of U.S. Cellular subscriber, in the format comprising: 1) 'tel:' protocol identifier, 2) country code of one preceded by '+', 3) ten digit CTN preceded by 1, for example, tel:+15087300001 The address must be must be URL-escaped where %3A represents ‘:’ and	No

		%2B represents '+'. Example: %3A%2B15087300001	
clientCorrelator	xsd:string	A correlator that the client can use to tag this particular resource representation during a request to create a resource on the server.	Yes
outboundMMSMessage	OutboundMMSMessage	Provides a summary of the MMS message, e.g. subject. **(See table 1.1 for details).**	No
receiptRequest	common:CallbackReference	It defines the notification endpoint and parameters that will be used to notify the application when the message has been delivered to terminal or if delivery is impossible. **(See table 1.2 for details).**	Yes
senderAddress		This is the URL-escaped end user ID. It is the address to whom a responding MMS may be sent. The address (CTN) of U.S. Cellular subscriber, in the format comprising: 1) 'tel:' protocol identifier, 2) country code of one preceded by '+', 3) ten digit CTN preceded by 1, for example, tel:+15087300001 The address must be must be URL-escaped where %3A represents ':' and %2B represents '+'. Example: %3A%2B15087300001	No
senderName	xsd:string	Name of the sender to appear on the user's terminal as the originator of the	Yes

		message.	

Table 1.1: Sending MMS from an Application - Request Parameters (outboundMMSMessage type)

Parameter	Type	Description	Optional
subject	xsd:string	This is the text you want to appear in the subject line of the MMS.	Yes
priority	MessagePriority	The priority of the message: default is Normal. **(See table 1.3 for details).**	Yes

Table 1.2: Sending MMS from an Application - Request Parameters (receiptRequest type)

Parameter	Type	Description	Optional
notifyURL	xsd:anyURI	This is the URL-escaped URL to which you want a notification of delivery sent.	Yes

Table 1.3: Enumeration - MessagePriority

Enumeration	Description
Default	Default message priority
Low	Low message priority
Normal	Normal message priority
High	High message priority

5.1.3 Response

```
HTTP/1.1 201 Created
Date: Thu, 31 Oct 2013 12:35:29 GMT
Server: Jetty(6.1.x)
Content-Type: application/json
breadcrumbId: ID-ase2-51909-1383117280141-17-7
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

```
Location:
http://developerportal.uscellular.com/services/sendMMS/2_0/messaging/outbound/12345/reques
ts/102300574

Host: developerportal.uscellular.com

Authorization: Basic QWVwb25hVfdghfdghdfghdfgWF6eHN3MjNl

Content-Length: 157

Keep-Alive: timeout=5, max=100

Connection: Keep-Alive

{

  "resourceReference" : {

    "resourceURL" :
"http://developerportal.uscellular.com/services/sendMMS/2_0/messaging/outbound/12345/requests/1023005
74"

  }

}
```

Table 2: Sending MMS from an Application - Response Parameters

Parameter	Type	Description	Optional
senderAddress	N/A	This is the URL-escaped end user ID. It is the address to whom a responding MMS may be sent. In the sample above, this is 'tel%3A %2B12345678'.	No
request ID	N/A	This is the request ID. In the sample above, this is 'req123'.	No

! The Location header field shows the URI of the created message, including the **senderAddress** and **request ID** in the path. You can append '/deliveryInfos' to this URI to query the delivery status (see 'Query Delivery Status of an MMS' below). For convenience this URI is also included in the response body as the **resourceURL** pair within the **resourceReference** object.

5.2 Query Delivery Status of an MMS

This section describes how to query the delivery status of an MMS.

5.2.1 Request

```
GET
https://developerportal.uscellular.com/services/sendMMS/2_0/messaging/outbound/12345/requests/102300574/deliveryInfos HTTP/1.1

Authorization: Basic QWVwb25hVfdghfdghdfghdfgWF6eHN3MjNl

Host: developerportal.uscellular.com

Connection: Keep-Alive

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

5.2.2 Request Parameters

Table 3: Query Delivery Status of an MMS Request Parameters

Parameter	Type	Description	Optional
requestID	N/A	This is the request ID which was returned when an MMS was sent from the Application (See Table 2).	No

5.2.3 Response

```
HTTP/1.1 200 OK

Date: Thu, 31 Oct 2013 12:35:41 GMT

Server: Jetty(6.1.x)

Content-Type: application/json

X-Forwarded-Server: developerportal.uscellular.com

Authorization: Basic QWVwb25hVfdghfdghdfghdfgWF6eHN3MjNl

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

Host: developerportal.uscellular.com
```

```
Content-Length: 289
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
{
  "deliveryInfoList" : {
    "resourceURL" :
"http://developerportal.uscellular.com/services/sendMMS/2_0/messaging/outbound/12345/reque
sts/102300574/deliveryInfos/",
    "deliveryInfo" : [ {
      "address" : "tel:+14167001003",
      "deliveryStatus" : "DeliveredToTerminal"
    } ]
  }
}
```

5.2.4 Response Parameters

The deliveryInfoList contains the delivery information for each address to which the message was sent. The information for each address is contained in a **deliveryInfo** array.

Table 4: Query Delivery Status of an MMS - Response Parameters (deliveryInfoList type)

Parameter	Type	Description	Optional
resourceURL	xsd:anyURI	This is a reference URL to the specific query.	No
link	common:Link[0..unbounded]	Linked to other resources that are in relationship with the resource.	Yes
deliveryInfo	DeliveryInfo[1… unbounded]	Lists the delivery information for an address.	No

Table 4.1: Query Delivery Status of an MMS - Response Parameters (deliveryInfo type)

Parameter	Type	Description	Optional
address	xsd:anyURI [1… unbounded]	Destination address(es) for the message.	No
deliveryStatus	DeliveryStatus	Indicates the delivery status. **(See table 4.2 for details).**	No

Table 4.2: Enumeration- deliveryStatus

Enumeration	Description
DeliveredToTerminal	Successful delivery to Terminal
DeliveryUncertain	Delivery status unknown, e.g. because it was handed off to another network.
DeliveryImpossible	Unsuccessful delivery; the message could not be delivered before it expired.
MessageWaiting	The message is still queued for delivery. This is a temporary state, pending transition to one of the preceding states
DeliveredToNetwork	Successful delivery to the network enabler responsible for routing the MMS.

5.3 Notify Client About Outbound Message Delivery Status

This section describes the format of delivery notifications sent from the server to the client application. A delivery notification is sent to the client application callback URL provided in the sendMMS request, typically when the message is DeliveredToTerminal or DeliveryImpossible.

Starting and stopping notifications is carried out via the Dev Portal. For instructions on how to do this, please see Appendix A.

5.3.1 Request

```
POST .../notifications/DeliveryInfoNotification/77777 HTTP/1.1
Accept: application/json
Content-Type: application/json; charset=UTF-8
Host: example.com:80

{"deliveryInfoNotification": {
        "deliveryInfo": [{
                "address": "tel:+19585550104",
                "deliveryStatus": "DeliveredToTerminal"
        }],
        "link": [{
"href": "http://developerportal.uscellular.com/sendMMS/2_0/messaging/outbound/tel%3A
%2B19585550100/requests/req123",
}]
   }
 }
```

5.3.2 Request Parameters

Table 5: notifyMessageDelivery Receipt - Request Parameters (deliveryInfoNotification type)

Parameter	Type	Description	Optional
deliveryInfo	DeliveryInfo[1 …unbounded]	Lists the delivery information for an address.	No
link	common:Link[0..unbounded]	Linked to other resources that are in relationship with the resource.	Yes

Table 5.1: notifyMessageDelivery Receipt - Request Parameters (deliveryInfo type)

Parameter	Type	Description	Optional

address	xsd:anyURI [1… unbounded]	Outbound message destination address.	No
deliveryStatus	DeliveryStatus	Indicates the delivery result for the destination address. **(See table 4.2 for details)**.	Yes

5.3.3 Response

```
HTTP/1.1 200 OK

Content-Type: application/json

Date: Thu, 04 Jun 2009 02:51:59 GMT
```

5.3.4 Response Parameters

N/A

6 Receiving MMS

6.1 Retrieve List of Messages Sent to an Application

This section describes how to retrieve a list of MMS messages sent to your application.

6.1.1 Request

```
GET
https://developerportal.uscellular.com/services/getReceivedMms/2_0/messaging/inbound/registrations/req123/messages?maxBatchSize=2 HTTP/1.1

Authorization: Basic QWVwb25hVfdghfdghdfghdfgWF6eHN3MjNl

Host: developerportal.uscellular.com
```

6.1.2 Request Parameters

Table 6: Retrieve List - Request Parameters

Parameter	Type	Description	Optional
registration ID	URI parameter	This is the registration ID agreed with the OneAPI operator. In the sample code above, this value is 'req123'.	No
maxBatchSize	xsd:int	This is the maximum number of messages to retrieve in this request.	Yes
retrievalOrder	retrievalOrder	The order is which you want the received MMS messages retrieved. Allowed values are 'OldestFirst or 'NewestFirst.	Yes
priority	messagePriority	The priority of the messages to poll from the gateway. All messages of the specified priority and higher will be retrieved. If not specified, all messages will be returned, i.e. the same as specifying Low. Allowed values	Yes

		are Default, Low, Normal or High.	

6.1.3 Response

```
HTTP/1.1 200 OK
Date: Thu, 31 Oct 2013 12:35:41 GMT
Server: Jetty(6.1.x)
Content-Type: application/json
X-Forwarded-Server: developerportal.uscellular.com
Authorization: Basic QWVwb25hVfdghfdghdfghdfgWF6eHN3MjNl
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
Host: developerportal.uscellular.com
Content-Length: 289
Keep-Alive: timeout=5, max=100
HTTP/1.1 200 OK
Content-Type: application/json
Date: Fri, 23 Sep 2011 10:00:29 GMT
Content-Length: 1075
{"inboundMessageList": {
"inboundMessage": [ {
"destinationAddress": "12345",
"senderAddress": "tel:+14567800001",
"dateTime": "2011-09-27T15:35:31.000+08:00",
"resourceURL": "http://developerportal.uscellular.com/services/getReceivedMms/2_0/messaging/inbound/registrations/req123/messages/1100020",
"link": {
"href":"http://developerportal.uscellular.com/services/getReceivedMms/2_0/messaging/inbound/registrations/req123/msg123/attachments/attach123",
"rel": "attachment"
},
"messageId": "1100020",
"inboundMMSMessage": {
"subject": “Sample MMX Notification”,
"priority": "HIGH",
"link": {
"href":"http://developerportal.uscellular.com/services/getReceivedMms/2_0/messaging/inbound/registrations/req123/msg123/attachments/attach123",
"rel": "attachment"
},
"bodyText": “See attached file”
},
}],
"totalNumberOfPendingMessages": 0,
"numberOfMessagesInThisBatch": 1,
"resourceURL": "http://developerportal.uscellular.com/services/getReceivedMms/2_0/messaging/inbound/registrations/req123/msg123/attachments/attach123"
}}
```

6.1.4 Response Parameters

Table 7: Retrieve List - Response Parameters (inboundMessageList Type)

Parameter	Type	Description	Optional
inboundMessage	inboundMessage [0..unbounded]	It may contain an array of messages received according to the specified registrationid. **(See table 7.1 for details)**.	Yes
totalNumberOfPendingMessages	xsd:int	This indicates the total number of pending messages awaiting retrieval from gateway storage.	Yes
numberOfMessagesInThisBatch	xsd:int	This indicates the number of messages in the response.	Yes
resourceURL	xsd:anyURI	This is a link to the message. Use this to retrieve the entire message including attachments (see the following section, 'Retrieve URIs of Media Attached to a	No

		Received MMS')	

Table 7.1: Retrieve List - Response Parameters (inboundMessage Type)

Parameter	Type	Description	Optional
destinationAddress	xsd:anyURI	The number associated with the service, e.g. an agreed short code.	No
senderAddress	xsd:anyURI	This is the MSISDN of the sender. Indicates message senderAddress.	No
dateTime	xsd:dateTime	The date and time at which the message was received.	Yes
resourceURL	xsd:anyURI	This is a link to the message. Use this to retrieve the entire message including attachments (see the following section, 'Retrieve URIs of Media Attached to a Received MMS')	Yes
link	common:Link[0..unbounded]	Link to other resources that are in relationship with the resource. The 'rel' parameter is a free string set by the server implementation. It's used to indicate a relationship between the current resource and an external resource. In this case, an 'attachment'.	Yes
messageId	xsd:string	This is a server-generated message identifier.	Yes
inboundMMSMessage	inboundMMSMessage	Inbound MMS Message **(See table 7.2 for details)**.	Choice

Table 7.2: Retrieve List - Response Parameters (inboundMMSMessage Type)

Parameter	Type	Description	Optional
subject	xsd:string	If present, indicates the subject of the received message.	Yes

priority	MessagePriority	The priority of the message: default is Normal. **(See table 1.3 for details)**.	Yes
link	common:Link	Link to individual attachment.	Yes
bodyText	xsd:string	Contains the message body if it is encoded as ASCII text	Yes

6.2 Retrieve URIs of Media Attached to a Received MMS

This section describes how to retrieve the URIs of media attached to MMS messages received by your application. Use the messageId in the response to Retrieve List of Messages identified by RegistrationId.

6.2.1 Request

```
GET
https://developerportal.uscellular.com/services/getReceivedMms/2_0/messaging/inbound/registrations/req123/messages/1000023?resFormat=JSON HTTP/1.1

Authorization: Basic QWVwb25hVfdghfdghdfghdfgWF6eHN3MjNl

Host: developerportal.uscellular.com
```

6.2.2 Request Parameters

Table 8: Retrieve URIs - Request Parameters

Parameter	Type	Description	Optional
registration ID	URI parameter	This is the registration ID agreed with the OneAPI operator. In the sample code above, this value is '1000018'.	No
messageId	URI parameter	This is the ID of the received MMS message. In the sample code above, this value is '1000023'.	No

resFormat	URI parameter	The response content type. For this method it is mandatory to set it to JSON.	No

6.2.3 Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT
{"inboundMessage": {
"destinationAddress": "12345",
"senderAddress": "tel:+14567800001",
"dateTime": "2013-09-29T16:05:08.000+08:00",
"resourceURL":
"https://developerportal.uscellular.com/services/getReceivedMms/2_0/messaging/inbound/registrations/req123/messages/1000023",
"messageId": "1000023",
"inboundSMSTextMessage": null,
"inboundMMSMessage": {
"subject": "subject",
"priority": "HIGH",
"link": [ {
"rel": "attachment",
"href": "https://developerportal.uscellular.com/OneMMXMO/1000023/1"
}],
"bodyText": "See attached picture"
}
}}
```

6.2.4 Response Parameters

Table 9: Retrieve URIs - Response Parameters (inboundMessage Type)

Parameter	Type	Description	Optional
destinationAddress	xsd:anyURI	The number associated with the service, e.g. an agreed short code.	No
senderAddress	xsd:anyURI	This is the number of the sender.	No
dateTime	xsd:dateTime	The date and time at which the message was received.	Yes
resourceURL	xsd:anyURI	This is a self-referring URL.	Yes
messageId	xsd:string	This is the ID of the MMS message containing the media. This parameter is normally optional but for MMS it is mandatory. It must be provided if the message has a type other than InboundSMSTextMessage.	No
inboundSMSTextMessage	InboundSMSTextMessage	Inbound SMS Text Message. Mandatory, if SMS text message is used.	Choice
inboundMMSMessage	inboundMMSMessage	Contains details of the inbound MMS message. Mandatory, if MMS message is used. **(See table 9.1 for details)**.	Choice
inboundIMMessage	inboundIMMessage	Inbound IM Message. Mandatory, if IM message is used.	Choice

Table 9.1: Retrieve URIs - Response Parameters (inboundMMSMessage Type)

Parameter	Type	Description	Optional
link	common:Link	This is a link to other resources that are in a relationship with the resource, i.e. this message. For example, an individual attachment. The 'rel' parameter is a free string set by the server implementation. It's used to indicate a	Yes

		relationship between the current resource and an external resource. In this case, an 'attachment'.	
subject	xsd:string	This is the text in the subject line of the MMS.	Yes
priority	MessagePriorit y	The priority of the message: default is Normal. **(See table 1.3 for details)**.	Yes
bodyText	xsd:string	The actual text of the received MMS.	Yes

6.3 Delete a Received MMS from Storage

This method uses the DELETE form of the same HTTP method and parameters as described in the section above, to *Retrieve URIs of Media Attached to a received MMS,* except that the *resFormat* URI parameter is not required.

6.4 Notify Client about Message Arrival

This section describes the format of MMS message arrival notifications sent from the server to the client application callback URL. The callback URL is configured for a given set of notification criteria (shortcode and keyword) in the Dev Portal.

Starting and stopping notifications is carried out via the Dev Portal. For instructions on how to do this, please see Appendix A.

The format of the message receipt notification sent to your **notifyURL** is:

```
{"inboundMessageNotification":{

"link":[],

"inboundMessage"{

"resourceURL":"https://developerportal.uscellular.com/services/getReceivedMms/2_0/messaging/
inbound/registrations/req123/messages/1000023",

"destinationAddress":"12345",

"senderAddress":"tel:+14567800001",

"dateTime":"2013-09-26T16:44:32.232+08:00",

"messageId":"1000023",

"link":[],

"inboundSMSTextMessage":null,
"inboundMMSMessage":{
"priority":"HIGH",
"subject":null,
"link":[],
"bodyText":null
},
"inboundIMMessage":null
}
}}
```

This will be sent for every MMS received (matching the optional criteria if provided).

The **inboundMessageNotification** object includes an **inboundMMSMessage** array of parameters.

Table 10: Receive Notification of a Received Message - Response Parameters (inboundMessageNotification Type)

Parameter	Type	Description	Optional
link	common:Link[0..unbounded]	Link to other resources, e.g. there can be a link to the subscription used to receive this message.	Yes
inboundMessage	inboundMessage	Details of the inbound message. **(See table 10.1 for details)**.	No

Table 10.1: Receive Notification of a Received Message - Response Parameters (inboundMessage Type)

Parameter	Type	Description	Optional
resourceURL	xsd:anyURI	Self referring URL. This will not be included in POST requests, MUST be included in responses to any HTTP method that returns an entity body, and in PUT requests.	Yes
destinationAddress	xsd:anyURI	The number associated with the service,	No

		e.g. an agreed short code.	
senderAddress	xsd:anyURI	This is the number of the sender.	No
dateTime	xsd:dateTime	The date and time at which the message was received.	Yes
messageId	xsd:string	This is a server-generated message identifier. This field MUST be present when the type of the message differs from a plain text SMS, i.e. the element in the choice below has a type other than InboundSMSTextMessage.	Yes
link	common:Link[0..unbounded]	Link to other resources that are in relationship with the resource.	Yes
inboundSMSTextMessage	InboundSMSTextMessage	Inbound SMS Text Message, mandatory for SMS text messages.	Choice
inboundMMSMessage	InboundMMSMessage	Inbound MMS Message. **(See table 9.1 for details)**. Mandatory for MMS messages.	Choice
inboundIMMessage	InboundIMMessage	Inbound IM Message. Mandatory for IM messages.	Choice

6.4.1 Response

The client application should return HTTP 204 – No Content.

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

6.4.2 Response Parameters

N/A

7 Response Codes & Exceptions

7.1 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 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.

7.2 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.

7.2.1 Service Exceptions

The following service exceptions may be thrown:

Table 11: 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.
SVC0007 – Invalid charging information	The charging information provided is invalid. Update your request and re-submit.
SVC0270 – Charge failed	The charge failed due to, for example, the transaction not being found or payment not allowed. Contact the support team.

7.2.2 Policy Exceptions

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

POL0001 - Policy error occurred

The above 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 12: 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-010: Subscriber target not authorized	Authorisation from the subscriber has not been obtained through the Consent service. Please request authorisation for this subscriber through the Consent Service.
POL-011: Charging not supported	Inline charging is not supported for this operator. Re-submit your request without charging information.
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.
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-020: Max Message Length is enforced, and max message length has been exceeded	A maximum message length policy is in place and you have exceeded this. Check you SLA for the maximum message length, update your message and re-submit your request.
POL-021: Min Message Length is enforced, and message length is less than min allowed	A minimum message length policy is in place and you have a message length that is less than this minimum. Check your SLA for the maximum message length, update your message and re-submit your request.
POL-022: Receipting is enforced, and receipting has not been enabled	A receipt has been requested but it is not enabled for this service. Remove the receipt request and re-submit your request.
POL-038: Max Charge Amount is enforced and maximum charge amount has been exceeded	A maximum charge amount is enforced and has been exceeded. Check your SLA for this limit and re-submit your request with the correct amount.
POL-039: Min Charge Amount is enforced and charge amount is less than minimum value	A minimum charge amount is enforced and a value less than this has been used. Check your SLA for this limit and re-submit your request with the correct amount.
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-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 Starting/Stopping Notifications

Starting and stopping notifications is managed from the Dev Portal. You will use one of the codes assigned to your Partner entity on the platform, or configure a callback URL, and keyword to automatically start receiving SMS arrival and receipt notifications. You can pause or stop by a button from the GUI.

The steps described below assume that you have a login on the Dev Portal which allows you to manage notifications for your Partner's applications.

A.1 Starting Notifications

The task is divided into three sub-sections in a standard workflow:

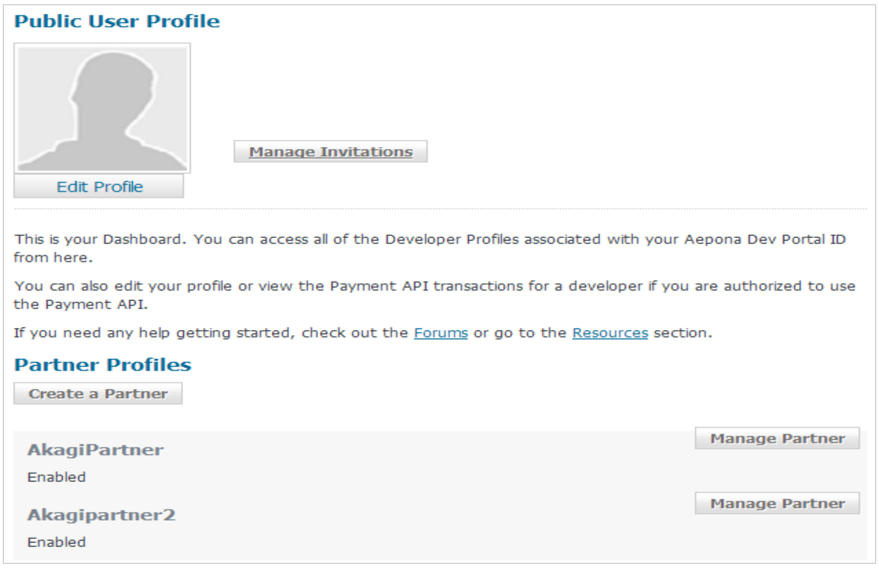
1. Requesting Codes
2. Creating Keywords
3. Creating Notifications

A.1.1 Requesting Codes

You need to request the administrator to assign a code to your project.

1. Login to the Dev Portal.

   The Dashboard is displayed, as shown below. Partners associated with you are displayed in the Partner Profiles area.



This is your Dashboard. You can access all of the Developer Profiles associated with your Aepona Dev Portal ID from here.

You can also edit your profile or view the Payment API transactions for a developer if you are authorized to use the Payment API.

If you need any help getting started, check out the Forums or go to the Resources section.

2 Click **Manage Partner** for the Partner whose application you want to set up the notification for.

The Partner Profile page is displayed. Registered applications are listed as projects in the Partner Projects area.

3 Click **Manage Project** for the relevant project.

The Project Profile page is displayed for your selected project.

4 Click **Manage Notifications**.

The Manage Notifications page is displayed.

5 Click **Request Number**.

The Create a New Number page is displayed.

6 Select the **Number Type** to request from the dropdown list: Short Code or Long Number.

7 Select to have a number assigned automatically by the Admin Portal administrator.

OR

to specify one for the project, by entering the **Number** in the field that will appear. This will be unique to your project, called an unshareable number.

8 Select the **Enabler Type**: SMS.

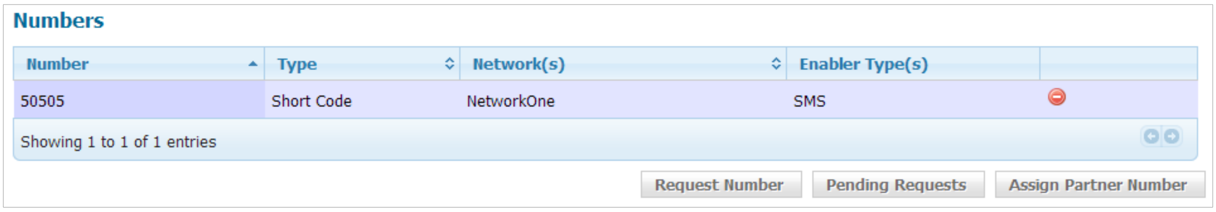
9 Select **Country** and **Networks** from the list displayed. You can assign only one network to a long number.

? If you are testing your application against the Sandbox version of the service, choose the Test network.

10 Click **Submit**.

Your request is submitted to the administrator, and until approved, will be listed in the Pending Requests page. (Click **Pending Requests** on the Manage Notifications page.)

You will receive an email when your request is approved. It will contain the number assigned. Your Manage Notifications page will show the assigned number in the Number panel as shown below:



Numbers

Number	Type	Network(s)	Enabler Type(s)	
50505	Short Code	NetworkOne	SMS	⊖

Showing 1 to 1 of 1 entries

? You can remove the number from your project by using the red icon in the Number panel. To add it back, use the **Assign Partner Number** button.

? As an option, you can create **Callback Credentials** from the Manage Notifications page. This will add authentication for the notification to access your application.

A.1.2 Creating Keywords

You need to create a keyword in combination with the number you are going to use for your notification. Each keyword+number pair must be unique across the platform, which means that if you are creating a single notification using your application specific number, keyword isn't necessary. In this case you can skips the steps below and go directly to create your notification.

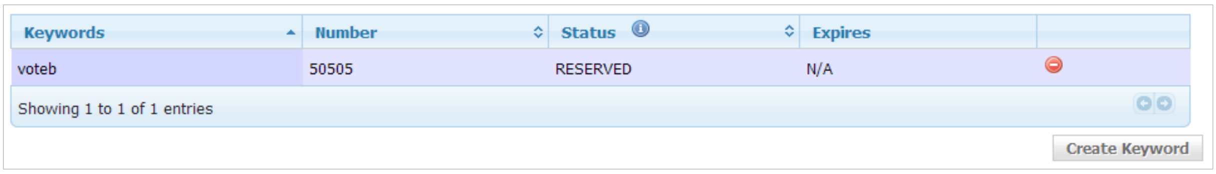
? A keyword can be created during notification creation, also.

1 From the Manage Notifications page, click **Create Keyword**. The Create Keyword page is displayed.

2 Specify the number you wish to associate the keyword with, from the **Number** dropdown list.

3 Enter the **Keyword**. An error message will be returned if the keyword is in use or it does not conform to the system rules.

Manage Notifications page is refreshed and shows the keyword created, in RESERVED status, as shown below:



Keywords	Number	Status	Expires	
voteb	50505	RESERVED	N/A	⊖

Showing 1 to 1 of 1 entries

The status will change to ACTIVE when the keyword is in use with a notification. The RESERVED status will expire after a system configured number of days, for example, 180 days. See Appendix A3 .

A.1.3 Creating Notifications

You are now ready to create a notification, which will start immediately.

1 From the Manage Notifications page, click **Create Notifications**.

2 Select the **Notification Type** from the dropdown list. SMSX is available by default.

3 Select the **Number** to use from the dropdown list.

4 Select the **Protocol** from the dropdown list.

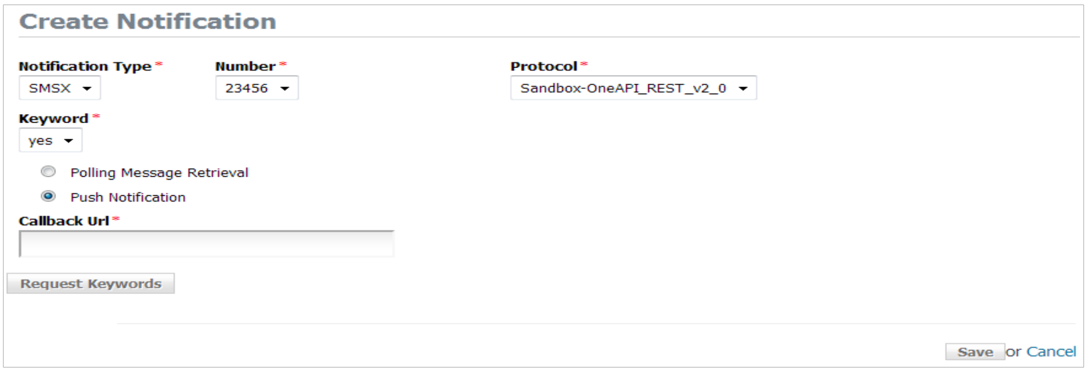
> ! If you are testing your application against the Sandbox version of the service, choose Sandbox-OneAPI_REST-v2_0.

5 Select the **Keyword** from the dropdown list, OR create one by clicking **Request Keywords**.

6 Select either Poll or Push notification model:

Notification Model	Description
Polling Message Retrieval	Cache & collect model. No application callback is required. The application will periodically retrieve messages from the platform via the API.
Push Notification	Direct push of notifications. An application callback is required.

7 If Push Notification is selected, enter the **Callback Url** in the field that appears:



Create Notification

Notification Type * SMSX

Number * 23456

Protocol * Sandbox-OneAPI_REST_v2_0

Keyword * yes

○ Polling Message Retrieval

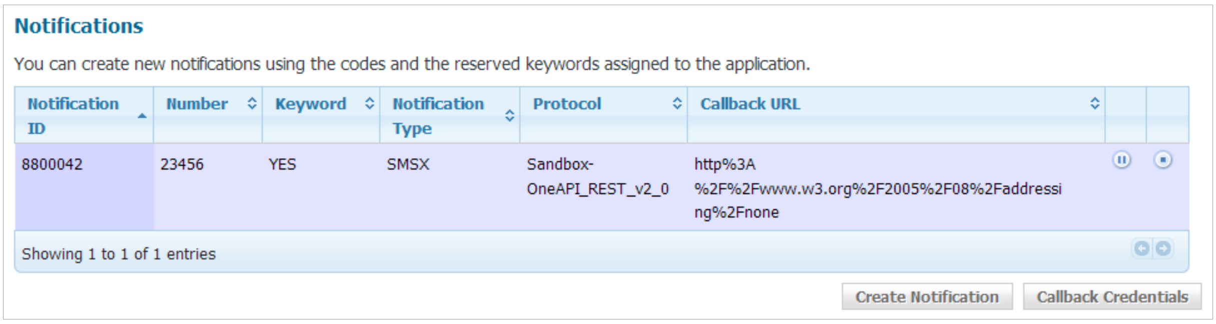
● Push Notification

Callback Url *

8 Click **Save**.

The notification is started.

The Manage Notifications page will be refreshed to display the notification with its ID, and the Callback URL as entered, with the pause and stop buttons at the right hand end of the row, as shown in the screenshot below:



Notifications

You can create new notifications using the codes and the reserved keywords assigned to the application.

Notification ID	Number	Keyword	Notification Type	Protocol	Callback URL		
8800042	23456	YES	SMSX	Sandbox-OneAPI_REST_v2_0	http%3A%2F%2Fwww.w3.org%2F2005%2F08%2Faddressing%2Fnone	⏸	■

Showing 1 to 1 of 1 entries

B Pausing, Restarting and Stopping Notifications

To pause, restart and stop a notification:

1 From the Dashboard, click **Manage Partner > Manage Project > Manage Notifications** to find the notification you wish to manage.

2 Click the appropriate button at the right end of the row:

- To pause a notification that has been started, click the ⏸ icon.
- To restart a notification that has been paused, click the ▶ icon.
- To stop a notification, click the ■ icon at the far right.

3 Confirm your action when prompted.

! When you stop a notification, it will be removed from the system. You will not be able to restart it, but you can recreate an identical notification.

C De-assigning and Reactivating Keywords

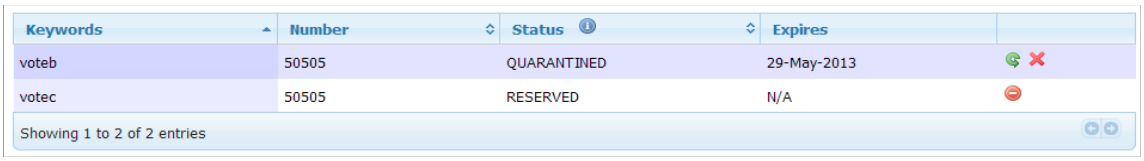
Keywords in RESERVED status can be de-assigned from the application and put into QUARANTINED status, and reactivated back to RESERVED status.

To de-assign a keyword from a project:

1 From the Manage Notifications page, find the keyword you wish to de-assign.

2 Click the ⊖ icon on the right.

The keyword is now in QUARANTINED status, as shown below, showing its expiry date after the system configured number of days (90, in the example below):



Keywords	Number	Status	Expires	
voteb	50505	QUARANTINED	29-May-2013	⟳ ✖
votec	50505	RESERVED	N/A	⊖

Showing 1 to 2 of 2 entries

- To reactivate the keyword, to RESERVED status, click the **green icon** on the right.
- To purge the keyword, click the **red cross icon** on the right.

D MMS Sandbox Service

The sandbox service replicates real U.S. Cellular Open API MMS 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.

Send MMS Sandbox Request Example

```
POST
https://developerportal.uscellular.com/services/sendMMSSandbox/2_0/messaging/outbound/123
45/requests HTTP/1.1

Content-Type: multipart/form-data;boundary="================123456==";

Authorization: Basic QWVwb25hVfdghfdghdfghdfgWF6eHN3MjNl

Content-Length: 698

Host: developerportal.uscellular.com

Connection: Keep-Alive

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


--================123456==

Content-Disposition: multipart/form-data; name=”root-fields”

Content-Type: application/json


{"outboundMessageRequest":

      {

    "address": ["tel:+14567800001"],

    "clientCorrelator": "564654251",

    "outboundMMSMessage": {"priority": "High", "subject": "FN-MMS-01"},
    "receiptRequest":{"notifyURL":"http://example.com/notifications/"},
    "senderAddress": "12345",
      "senderName": "kieran"
     }
}
--================123456==
Content-Disposition: multipart/form-data; name=”attachments”
Content-Type: multipart/mixed; boundary=”===12345===”

--===12345===
Content-Disposition: attachment; filename="dublin_bay.jpg"
Content-Type: image/jpeg
--===12345===--
--================123456==--
```

Query MMS Delivery Status request:

```
GET
https://developerportal.uscellular.com/services/sendMMSSandbox/2_0/messaging/outbound/12345/requests/66100104/deliveryInfos HTTP/1.1
Authorization: Basic QWVwb25hVfdghfdghdfghdfgWF6eHN3MjNl
Host: developerportal.uscellular.com
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

End of Document