

Service Exposure OneAPI SMS Development

Ericsson Composition Engine

USER GUIDE

Copyright

© Ericsson AB 2012–2014. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

Trademark List

All trademarks mentioned herein are the property of their respective owners. These are shown in the document Trademark Information.



Contents

1	About This Document	1
1.1	Revision Information	1
1.2	Purpose	1
1.3	Target Group	1
1.4	Prerequisites	2
2	OneAPI SMS Overview	3
3	Security and Authentication	5
4	Interface: Sending SMS	7
4.1	Operation: Sending SMS from Web Application	7
4.2	Operation: Query SMS Delivery Status	11
5	Interface: Receiving SMS	13
5.1	Operation: Retrieving SMS Sent to Web Application	13
5.2	Operation: Subscribe to SMS Delivery Notifications	14
5.3	Operation: Stop Subscription to Delivery Notifications	17
6	Response Codes and Exceptions	19
6.1	Response Codes	19
6.2	Exceptions	19
	Glossary	25
	Reference List	27





1 About This Document

This chapter contains the following parts:

- Revision information
- Purpose
- Target audience
- Prerequisites

1.1 Revision Information

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

Table 1 Revision Information

Revision	Section	Change
G	Section 3 on page 5	Added information about the Access Key header in B2B scenario.
	Section 3 on page 5	Corrected authentication parameters in Table 2.
	Section 6.2.3 on page 22	Added reference to the integration adapter error codes in <i>Ericsson Composition Engine, System Administration Guide</i> , Reference [3].
H	Section 3 on page 5	Updated the description about OAuth authentication.
I	Section 4.1.2 on page 7	Updated the parameter descriptions of <code>message</code> .

1.2 Purpose

This document describes how to develop an application which is compatible with Service Exposure OneAPI SMS interface. The full specification of OneAPI SMS interface and examples with detailed explanation are provided in this document.

1.3 Target Group

This document is intended for application developers.



1.4 Prerequisites

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

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



2 OneAPI SMS Overview

The OneAPI SMS API of ECE is implemented based on the **SMS RESTful NetAPI** protocol. The OneAPI SMS API supports the following interfaces for the application to send and receive SMS:

- **Sending SMS**
 - Send an SMS to the mobile terminals
 - Query the delivery status of an SMS that is sent to the mobile terminals
- **Receiving SMS**
 - Retrieve an SMS that is sent from the mobile terminals
 - Subscribe to the delivery notifications of SMS that is sent to the applications
 - Stop the subscription to the delivery notifications of SMS that is sent to the applications

When the application sends the HTTP requests to the ECE, it is required to use UTF-8 encoding.

For the **Sending SMS** interface, the application is allowed to send the following SMS messages to the mobile terminals:

- A single text message
- A single binary message
- Segmented text messages
- Segmented binary messages

For descriptions about the **SMS RESTful NetAPI** protocol, refer to <http://oneapi.gsma.com/sms-restful-netapi/>.





3 Security and Authentication

A server-side certificate is used to secure the HTTPS connection and to confirm the server identity. `notifyURL` for Delivery Report supports both HTTP and HTTPS.

The client is authenticated with HTTP basic authentication or with OAuth authentication.

HTTP Basic Authentication

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

```
POST /oneapi/sms/1/inbound/subscriptions HTTP/1.0
Authorization: Basic cGFydG51cjFAYXBwMTphdXRob2s=
Content-Type: application/json
Content-Length: 246
Host: 127.0.0.1:38080
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.2 (java 1.5)
{"subscription":{
  "callbackReference":{
    "callbackData":"doSomething()",
    "notifyURL":"http://127.0.0.1:23001/smsReception/1375960474878"},
    "criteria":"Vote",
    "destinationAddress":"tel:123456",
    "notificationFormat":"JSON",
    "clientCorrelator":"123456026"
  }
}
```

The format of the Authorization header is as follows:

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

The following parameters are used in the authentication.

Table 2 Authentication Parameters

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

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

```
Authorization: Basic cGFydG51cjFAYXBwMTphdXRob2s=
```

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



OAuth Authentication

For OAuth authentication, only the client credential flow is supported for the client to get the access token. The OAuth authorization flow which uses authorization code to get access token is not supported.

For more information about client integration, see *Authorization Integration Guide*, Reference [1].

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

```
POST /oneapi/sms/smsmessaging/v1/outbound/tel:+10086/requests
Authorization: Bearer MkdPhwf2FCVm5rijcaCi
Host: ece.example.com:38080
Accept: application/json
Content-Type: application/json
{"outboundSMSMessageRequest":{
  "senderAddress":"tel:+10086",
  "outboundSMSTextMessage":{"message":"Hello World!"},
  "address":["tel:+8613500000000"],
  "receiptRequest":{
    "notifyURL":"https://app.example.com:7464/smsStatus",
    "callbackData":"doSomething()"
  }
}}
```

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

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

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

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

```
POST /oneapi/sms/smsmessaging/v1/outbound/tel:+10086/requests
Authorization: Bearer MkdPhwf2FCVm5rijcaCi
AccessKey: 1316kj
Host: ece.example.com:38080
Accept: application/json
Content-Type: application/json
{"outboundSMSMessageRequest":{
  "senderAddress":"tel:+10086",
  "outboundSMSTextMessage":{"message":"Hello World!"},
  "address":["tel:+8613500000000"],
  "receiptRequest":{
    "notifyURL":"https://app.example.com:7464/smsStatus",
    "callbackData":"doSomething()"
  }
}}
```



4 Interface: Sending SMS

The **Sending SMS** interface supports the application to send the following HTTP requests to the ECE:

- Send an SMS to the mobile terminals
- Query the delivery status of an SMS that is sent to the mobile terminals

4.1 Operation: Sending SMS from Web Application

The **Sending SMS** interface supports the application to send the following SMS messages to one or more mobile terminals:

- A single text message
- A single binary message
- Segmented text messages
- Segmented binary messages

4.1.1 Request

The following table describes the HTTP request to send an SMS message to one or more mobile terminals.

Table 3 Request of Sending SMS

URL Pattern	https://<serverRoot>/oneapi/sms/1/outbound/<senderAddress>/requests/ (1)
HTTP Method	POST
Content Type	application/json
<p>Example:</p> <pre>POST /oneapi/sms/1/outbound/tel:+10086/requests Authorization: Bearer MkdPhwf2FCVm5rijcaCi Host: example.com:38080 Accept: application/json Content-Type: application/json {"outboundSMSMessageRequest":{ "senderAddress":"tel:+10086", "outboundSMSTextMessage":{"message":"Hello World!"}, "address":["tel:+8613500000000"], "receiptRequest":{ "notifyURL":"https://example.com:7464/smsStatus", "callbackData":"doSomething()" } }}</pre>	

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



4.1.2 Request Parameters

The following table shows the parameters for sending an SMS request, which is of type `outboundSMSMessageRequest`.

Table 4 Request Parameters of Sending SMS

Parameter	Location	Type	Description	Mandatory
<code>senderAddress</code>	URL and payload	String	It is the address to which a responding SMS is sent. <code>senderAddress</code> in URL and payload must be same.	Yes
<code>address</code>	Payload	String	The phone number in <code>tel</code> URI format. Only global number is supported. “tel:” scheme and “+” identifier must be given. For example, <code>tel:+8613500000000</code> At least one address must be provided. ACR is not supported. The Maximum number of address is limited by SLA <code>MaxNumberOfRecipients</code>	Yes



Parameter	Location	Type	Description	Mandatory
message	Payload	String	<p>The message content is provided within the <code>outboundSMSTextMessage</code> element.</p> <p>The maximum length of a complete message is limited by the SLA parameters <code>MaxTextMessageSize</code> for text messages and <code>MaxBinaryMessageSize</code> for binary messages. To support the function of sending segmented messages, it is required to set the two parameters to larger values. Otherwise, ECE rejects the messages that are longer than the SLA values.</p> <p>Use the following formats for text and binary messages.</p> <ul style="list-style-type: none"> "message": "<content>": For sending a single text message. "message": "<binary>;<content>": For sending a single binary message. The <content> of the message is required to be in hexadecimal format with upper cases. "message": "<text><UDH 1>;<content 1>;<UDH 2>;<content 2>;<UDH n>;<content n>": For sending segmented text messages. Each piece of <UDH> + <content> is required to be in hexadecimal format with upper cases and no longer than 280 bytes. "message": "<binary><UDH 1>;<content 1>;<UDH 2>;<content 2>;<UDH n>;<content n>": For sending segmented binary messages. Each piece of <UDH> + <content> is required to be in hexadecimal format with upper cases and no longer than 280 bytes. 	Yes
<p>Examples of the message parameter:</p> <p><i>Sending a single text SMS:</i></p> <pre>"outboundSMSTextMessage": {"message": "Hello World!"}</pre> <p><i>Sending a single binary SMS:</i></p> <pre>"outboundSMSTextMessage": {"message": "<binary>;313233343536373839"},</pre> <p><i>Sending Segmented text messages:</i></p> <pre>"outboundSMSTextMessage": {"message": "<text>050003120301;68656C6C6F;050003120302;676F6F64206D6F726E696E67"},</pre> <p><i>Sending Segmented binary messages:</i></p> <pre>"outboundSMSTextMessage": {"message": "<binary>050003120201;31323334;050003120202;3536373839"},</pre>				
clientCorrelator	Payload	String	It uniquely identifies the request.	No
senderName	Payload	String	It is the URL-escaped name of the sender to appear on the terminal.	No



Parameter	Location	Type	Description	Mandatory
notifyURL	Payload	URI	It is the URL to which a notification of delivery is sent. If notifications are required, include the notifyURL parameter within the receiptRequest element.	No
callbackData	Payload	String	It is passed back in the notification. It can be used to identify the message that the receipt relates to (or any other useful data, such as a function name). It is only valid if notifications are required. When it is valid, include the callbackData parameter with the notifyURL parameter within the receiptRequest element.	No

4.1.3 Response

The following table shows the parameter of the response to the Sending SMS request from the application.

Table 5 Response Parameter for Sending SMS

Parameter	Description
Location	The header field shows the URI of the created message, including senderAddress (For example, tel:+10086) and requestID (For example: 10c3b83da7040bfd@138e1821414).

An example response:

```
201 Created
Content-Type: application/json
Location: https://ece.example.com:38080/oneapi/sms/1/outbound/
tel:+10086/requests/10c3b83da7040bfd@138e1821414
Date: Wed, 01 Aug 2012 09:26:34 GMT
Content-Length: 0
```

4.1.4 Delivery Notification Sent to NotifyURL

Delivery notification is sent for every SMS that is created, if notifyURL is provided. Following is an example of the delivery notification.

```
POST /smsStatus
Host: ece.example.com:38080
Content-Type: application/json

{"deliveryInfoNotification":{
  "callbackData":"doSomething()",
  "deliveryInfo":[{
    "address":"tel:+8613500000000",
    "deliveryStatus":"DeliveredToTerminal"
  }]
}}
```

The deliveryStatus pair in the deliveryInfo object can take the values in Table 9, except for MessageWaiting which is the initial status. The



`callbackData` is also passed back, echoing what is provided when creating a message to send.

The following is an example of the Delivery Report response:

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Length: 0
Date: Thu, 22 Aug 2013 09:00:12 GMT
```

4.2 Operation: Query SMS Delivery Status

The **Sending SMS** interface can be invoked to query the delivery status of an SMS that is sent from the web application.

4.2.1 Request

The following table describes the HTTP request to query the SMS delivery status identified by `requestId`.

Table 6 Request of Querying SMS Delivery Status

URL Pattern	<code>https://<serverRoot>/oneapi/sms/1/outbound/<senderAddress>/requests/<requestId>/deliveryInfos⁽¹⁾</code>
HTTP Method	GET
Content Type	application/json
Example: <pre>GET /oneapi/sms/1/outbound/tel:+10086/requests/ 10c0d78cd4bf77ee@138a3d02644/deliveryInfos Authorization: Bearer MkdPhwf2FCVm5rijcaCi Host: ece.example.com:38080 Accept: application/json</pre>	

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

4.2.2 Request Parameters

The following table shows the request parameters of querying the SMS delivery status.

**Table 7 Request Parameters of Querying SMS Delivery Status**

Parameter	Location	Type	Description	Mandatory
senderAddress	URL	String	It is the address to which a responding SMS is sent.	Yes
requestID	URL	String	It is returned from the Sending SMS request.	Yes

4.2.3

Response

The following table shows the parameter of the response to querying the SMS delivery status.

Table 8 Response Parameter of Querying SMS Delivery Status

Parameter	Description
deliveryInfoList	It is the delivery status response. It contains the delivery information for each address where the message is sent. The delivery information is in a <code>deliveryInfo</code> array. The <code>deliveryStatus</code> value is one of the values in Table 9.

An example response:

```
200 OK
Content-Type: application/json
Date: Wed, 01 Aug 2012 09:26:22 GMT

{"deliveryInfoList":{
  "resourceURL":"https://ece.example.com:38080/oneapi/sms/1/outbound
/tel:+10086/requests/10c3b83da7040bf@138e181e104/deliveryInfos",
  "deliveryInfo":{
    "address":"tel:+8613500000000",
    "deliveryStatus":"DeliveredToTerminal"
  }
}}
```

The following table lists the available delivery status.

Table 9 Delivery Status

Status	Description
DeliveredToTerminal	The message is delivered to terminal successfully.
DeliveryUncertain	The delivery status is unknown. For example, because the message was handed off to another network.
DeliveryImpossible	The message fails to be delivered. The message is not delivered before it expired.
MessageWaiting	The message is still queued for delivery. This status is temporary.
DeliveredToNetwork	The message is delivered to the network enabler that is responsible for routing the SMS.



5 Interface: Receiving SMS

The **Receiving SMS** interface supports the application to send the following HTTP requests to the ECE:

- Retrieve an SMS that is sent from the mobile terminals
- Subscribe to the SMS delivery notifications
- Stop the subscription to the SMS delivery notifications

5.1 Operation: Retrieving SMS Sent to Web Application

The **Receiving SMS** interface can be used to retrieve the SMS that is sent to web application.

5.1.1 Request

The following table describes the HTTP request to retrieve the SMS that is sent to the application identified by `registrationId`.

Table 10 Request of Retrieving an SMS Sent to Web Applications

URL Pattern	<code>https://<serverRoot>/oneapi/sms/1/inbound/registrations/<registrationId>/messages⁽¹⁾</code>
HTTP Method	GET
Content Type	application/json
<p>Example:</p> <pre>GET /oneapi/sms/1/inbound/registrations/12345/ messages?maxBatchSize=2 Authorization: Bearer MkdPhwf2FCVm5rijcaCi Accept: application/json Host: ece.example.com:38080</pre>	

(1) Replace `<serverRoot>` with the hostname of the OneAPI server that is being accessed.

5.1.2 Request Parameters

The following table shows the parameters of the request to retrieve an SMS that is sent to the application.

**Table 11** Request Parameters of Retrieving an SMS Sent to the Application

Parameter	Location	Type	Description	Mandatory
registrationID	URL	String	It is typically a short-code or virtual MSISDN that is agreed with the mobile operator for receipt of SMS messages.	Yes
maxBatchSize	URL	Integer	It is the maximum number of messages to be retrieved in the request. The actual maximum batch size is also limited by SLA MaxReturnedSMS. System chooses the smaller value from the value of maxBatchSize and MaxReturnedSMS.	No

5.1.3 Response

The following table shows the parameters of the response to retrieve an SMS sent to the web applications.

Table 12 Response Parameters of Retrieving an SMS Sent to the Applications

Parameter	Description
dateTime	The time when the message is received.
destinationAddress	The number associated with service (for example, an agreed short code)
message	The SMS message itself.
senderAddress	The MSISDN or Anonymous Customer Reference of the sender
numberOfMessagesInThisBatch	The number of messages in the batch.
resourceURL	self-referring resource URL

An example response:

```
200 OK
Content-Type: application/json
Date: Wed, 01 Aug 2012 09:26:22 GMT

{"inboundSMSMessageList":{
  "inboundSMSMessage":{
    "dateTime":"2012-08-01T09:25:55.815+0000",
    "destinationAddress":"tel:12345",
    "message":"Vote SMSes",
    "senderAddress":"tel:+8613500000000",
  },
  "numberOfMessagesInThisBatch":1,
  "resourceURL":"https://ece.example.com:38080/oneapi/sms/1/inbound/registrations/123456024/messages"
}
```

5.2 Operation: Subscribe to SMS Delivery Notifications

The **Receiving SMS** interface can be invoked to subscribe to SMS delivery notifications.



5.2.1 Request

The following table describes the HTTP request to subscribe to the delivery notifications for SMS that is sent to the applications.

Table 13 Request of Subscribing to the SMS Delivery Notifications

URL Pattern	<code>https://<serverRoot>/oneapi/sms/1/inbound/subscriptions</code> (1)
HTTP Method	POST
Content Type	application/json
<p>Example:</p> <pre> POST /oneapi/sms/1/inbound/subscriptions Authorization: Bearer MkdPhwf2FCVm5rijcaCi Accept: application/json Content-Type: application/json Host: ece.example.com:38080 {"subscription": { "callbackReference": { "callbackData": "doSomething()", "notifyURL": "https://example.com/notifications/ smsReception/1345198652101" }, "criteria": "Vote", "destinationAddress": "tel:12345", "notificationFormat" : "JSON", "clientCorrelator" : "123456024" } } </pre>	

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

5.2.2 Request Parameters

The following table shows the parameters of the request to subscribe to SMS delivery notifications.

Table 14 Request Parameters of Subscribing to SMS Delivery Notifications

Parameter	Location	Type	Description	Mandatory
callbackData	Payload	String	It is included in the notifyURL	No
clientCorrelator	Payload	String	It uniquely identifies the request. If the communication fails during the request, the request needs to be retried. The same clientCorrelator is used to avoid creating a duplicate subscription.	No



Parameter	Location	Type	Description	Mandatory
criteria	Payload	String	It is a case-insensitive text that is matched against the first word of the message, ignoring any leading white space. This allows to reuse a short code among various applications, each of which can register their own subscription with different criteria.	No
destinationAddress	Payload	String	It is the MSISDN, or short code agreed with the operator, with which people send an SMS to the application.	Yes
notificationFormat	Payload	String	It is the content type in which notifications are sent. It only supports JSON.	No
notifyURL	Payload	URI	It is the URI of the application to which notifications are sent.	Yes

5.2.3

Response

The following table shows the parameter of the response to subscribe to the SMS delivery notifications:

Table 15 Response Parameter of Subscribing to SMS Delivery Notifications

Parameter	Description
Location	The header field shows the URI that is used for getting the subscription. This parameter is mandatory.

An example response:

```
201 Created
Content-Type: application/json
Location:https://ece.example.com:38080/oneapi/sms/1/inbound/subscriptions/123456024
Date: Wed, 01 Aug 2012 09:25:55 GMT
Content-Length: 0
```

5.2.4

Message Receipt Notification Sent to NotifyURL

Message receipt notification can be sent for every received SMS (that match the optional criteria if provided).

The following is an example of message receipt notifications.

```
POST /notifications/smsReception/1345198652101
Content-Type: application/json
Host: ece.example.com:38080

{"inboundSMSMessageNotification":{
  "callbackData":"doSomething()",
  "inboundSMSMessage":{
    "dateTime":"2012-08-01T09:25:55.815+0000",
    "destinationAddress":"tel:+8613500000000",
    "message":"Vote SMSes",
    "senderAddress":"tel:111111"
  }
}}
```



The `inboundMessageNotification` object includes any `callbackData` and an `inboundSMSMessage` array. The description of `callbackData` and `inboundSMSMessage` array are same as Table 12.

5.3 Operation: Stop Subscription to Delivery Notifications

The Receiving SMS interface can be invoked to stop subscription to delivery notifications.

5.3.1 Request

The following table describes the HTTP request to stop subscribing to delivery notifications for SMS that is sent to the web applications.

Table 16 Request of Stopping Subscription to SMS Delivery Notifications

URL Pattern	<code>https://<serverRoot>/oneapi/sms/1/inbound/subscriptions</code> (1)
HTTP Method	DELETE
Content Type	application/json
<p>Example:</p> <pre>DELETE /oneapi/sms/1/inbound/subscriptions/123456021 Authorization: Bearer MkdPhwf2FCVm5rijcaCi Accept: application/json Host: ece.example.com:38080</pre>	

(1) Replace `<serverRoot>` with the hostname of the OneAPI server that is being accessed.

5.3.2 Request Parameters

The following table shows the parameter of the request to stop the subscriptions to SMS delivery notifications.

Table 17 Request Parameter of Stopping Subscriptions to SMS Delivery Notifications

Parameter	Location	Type	Description	Mandatory
<code>subscriptionID</code>	URL	String	It is created when the subscription of delivery notification is created.	Yes

5.3.3 Response

Following is an example of the response to stopping the subscription to delivery notifications.



```
204 No content
Content-Type: application/json
Date: Wed, 01 Aug 2012 09:23:43 GMT
Content-Length: 0
```



6 Response Codes and Exceptions

This chapter describes some response codes and exceptions.

6.1 Response Codes

The following table shows some response codes and their indications.

Table 18 Response Codes

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

6.2 Exceptions

Following is an example of exception.

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

{"requestError": {
  "policyException": {
    "messageId": "POL0001",
    "text": "A policy error occurred. Error code is maxBatchSize exceeded.
The maximum allowed maxBatchSize is %1.",
    "variables": "20" }
}}
```

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

Table 19 Parameters Description

Parameter	Description
400	Error code



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

6.2.1 Service Exceptions

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

Table 20 Service Exceptions

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



ID	Exception Text	Variables	HTTP Code
SVC1000	No resources. This exception is thrown if there are no server resources available to process the request.	None	503 Service unavailable
SVC0280	Message too long. Maximum length is %1 characters	%1 – number of characters allowed in a message.	400 Bad Request
SVC0283	Delivery receipt notification not supported	None	400 Bad Request

(1) For a reference to the error codes, see Section 6.2.3 on page 22.

6.2.2 Policy Exceptions

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

Table 21 Policy Exceptions

ID	Exception Text	Variables	HTTP Code
POL0001	A policy error occurred. Error code is %1.	%1 – explanation of the error ⁽¹⁾	403 Forbidden
POL0002	Privacy verification fails for address %1, request is refused	%1 – address privacy verification fails for	403 Forbidden
POL0003	Too many addresses specified in message part %1.	%1 – message part	403 Forbidden
POL0004	Unlimited notification request not supported	None	403 Forbidden
POL0005	Too many notifications requested	None	403 Forbidden
POL0006	Group specified in message part %1 not allowed. ⁽²⁾	%1 – message part	403 Forbidden
POL0007	Nested groups specified in message part %1 not allowed. ⁽³⁾	%1 – message part.	403 Forbidden
POL0008	Charging is not supported	None	403 Forbidden
POL0009	Invalid frequency requested	None	403 Forbidden



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

(1) For a reference to the error codes, see Section 6.2.3 on page 22.

(2) Group means an address that refers to more than one end user.

(3) Group means an address which refers to more than one end user. Groups cannot contain addresses which are themselves groups.

6.2.3

Error Codes

This section gives a reference to the error codes in service exception SVC0001 in Table 20 and in policy exception POL0001 in Table 21.

Table 22 Error Code Reference

Error Code	Indication
1	Unexpected network or system error
2	Request time-out
3	The service capability is inactive
4	Cannot connect to rule engine



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



Error Code	Indication
71	The request sender or the group owner belongs to the service provider, for ParlayX ALM
72	The request sender is not authorized in the southbound resource
80	Too many subscriptions for a watcher or a presentity (ParlayX Presence)
81	The subscription is expired or does not exist
100	Failed to deliver to all addresses
101	Internal database error
102	Internal system congestion error
103	Message short code error
104	Message keyword error
110	Database integrity constraint violation
200	Duplicated session
300	MSISDN not allowed
400	The authorization scope is not sufficient
401	Authentication failure
402	HTTP failure
9000-9499	Integration adapter service exception SVC0001 error codes. For more information, see <i>Ericsson Composition Engine, System Administration Guide</i> , Reference [3].
9500-9999	Integration adapter policy exception POL0001 error codes. For more information, see <i>Ericsson Composition Engine, System Administration Guide</i> , Reference [3].



Glossary

API

Application Programming Interface

B2B

Business to Business

ECE

Ericsson Composition Engine

HTTP

Hypertext Transfer Protocol

HTTPS

HTTP over SSL

ID

Identification

OAuth

Open standard for Authorization

OneAPI

Open Network Enablers API

REST

Representational State Transfer

SLA

Service Level Agreement

SMS

Short Message Service

URI

Uniform Resource Identifier

URL

Uniform Resource Locator





Reference List

Ericsson Documents

- [1] *Authorization Integration Guide, 2/1553-CXP 904 0266 Uen*
- [2] *Ericsson Composition Engine, Glossary, 0033-HSC 901 024/1 Uen*
- [3] *Ericsson Composition Engine, System Administration Guide, 1543-APR 901 0383/30 Uen*