SDP Solution

[API Reference (SMS, ParlayREST)]

Ethio telecom

Huawei Technologies Co., Ltd

Contents

1 Overview

1.1 API Functions

The SDP provides SMS capability application programming interfaces (APIs) for third-party applications (App for short) to connect to it and use its SMS capability to send and receive SMS messages. The App is generally developed by various partners of the SDP.

NOTE

Partners are the enterprises and individuals who sign a contract and cooperate with carriers in utilizing the SDP. Partners include SPs, CPs, Developers, and Enterprises. In this document, partners are mainly the SPs, Developers, and Enterprises who use APIs for secondary development.

Table 1-1 describes functions of SMS capability APIs provided by the SDP.

Table 1-1 Functions of SMS capability APIs

Function	Subfunction	Description	API
Receiving SMS messages	Receiving SMS messages in Notify mode	The App (functioning as the client) invokes the Inbound SMS message subscriptions API to subscribe to mobile originated (MO) SMS message notification on theSDP (functioning as the server). When receiving an MO SMS message from a user, the SDP (functioning as the client) invokes the Client notification about inbound SMS message API to send the MO SMS message to the App (functioning as the server). Before the App is brought offline, the App (functioning as the client) invokes the Individual inbound SMS message subscription API to unsubscribe from MO SMS message notification on the SDP (functioning as the server).	• 2.2 Inbound SMS message subscriptions • 2.3 Client notification about inbound SMS message • 2.4 Individual inbound SMS message subscription
	Receiving SMS messages in Get mode	The App (functioning as the client) invokes the Inbound SMS messages Retrieve and Delete using registration API to obtain MO SMS messages from the SDP (functioning	2.5 Inbound SMS messages Retrieve and Delete using

Function	Subfunction	Description	API
		as the server).	registration
Sending SMS messages	The App sends SMS messages to users and receives status reports from the SDP in Notify mode.	The App (functioning as the client) invokes the Outbound SMS message delivery notification subscriptions API to send SMS messages through the SDP (functioning as the server).	• 3.2 Outbound SMS message delivery notification subscriptions • 3.3 Outbound SMS message requests • 3.4 Client notification SMS message delivery status • 3.4 Client notification SMS message delivery status
		After the App sends an SMS message to a user, the SMSC sends a status report to the SDP. The SDP (functioning as the client) then invokes the Client notification SMS message delivery status API to send the status report to the App (functioning as the server).	
	Receiving status reports in Get mode	After the App sends an SMS message to a user, the SMSC sends a status report to the SDP. The App (functioning as the client) then invokes the Outbound SMS message delivery status API to obtain the status report from the SDP (functioning as the server).	• 3.3 Outbound SMS message requests • 3.6 Outbound SMS message delivery status

The App receives MO SMS messages and status reports from the SDP in either of the following modes:

- Notify

  The SDP notifies the App of MO SMS messages and status reports immediately when receiving them from users. To use this mode, the App must subscribe to MO SMS message notification on the SDP. This mode has the following features:

  - Real-time

    The App receives users' SMS messages and status reports in real time, which provides a pleasant user experience in interactive services.

  - High requirements for hardware

    If the App hardware performance does not match that of the SDP, the App may fail to process surging requests in a timely manner.

- Get

  The App periodically obtains MO SMS messages and status reports from the SDP. This mode has the following features:

  - Non-real-time

    The App cannot receive users' SMS messages and status reports in real time, which degrades user experience in interactive services.

  - Low requirements for hardware

Partners can select a mode based on the service or application requirements. The Notify mode is recommended. Partners must use the Notify mode to receive messages involved in an on-demand SMS service or application.

1.2 Level of Requirement for Parameters

The App must develop APIs based on the level of requirement for each parameter.

Table 1-2 Level of requirement for parameters

Type	Description
Mandatory	A parameter is always mandatory in a request. Parameters with the **Mandatory** requirement are used for access authentication or service processing. If a parameter with the **Mandatory** requirement is left empty in a request, access authentication or service processing fails and the request fails.
Conditional	A parameter is mandatory or optional in specified conditions. Parameters with the **Conditional** requirement are used for access authentication or service processing in specified conditions. If the specified conditions are met but a parameter with the **Conditional** requirement is left empty in a request, access authentication or service processing fails and the request fails.
Optional	A parameter is always optional. Parameters with the **Optional** requirement are not used for service processing.

1.3 Request Format

A ParlayREST request provided by the SDP consists of the request line, message header, and message body.

The SDP provides the ParlayREST request in the following format:

```
POST /1/smsmessaging/outbound/861234/requests HTTP/1.1
Authorization: WSSE realm="SDP",profile="UsernameToken"
X-WSSE: UsernameToken
Username="35000001",PasswordDigest="RnafRPVpUfWp7I2IQnHZxkjh+fc=",Nonce="201304271
8194900001", Created="2013-04-27T18:19:49Z"
X-RequestHeader: request
ServiceId="3500001000001",LinkId="12132131",FA="tel:12312312123"
Accept-Encoding: gzip,deflate
Accept: application/xml
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.135.178.84:9088
Content-Type: application/xml; charset=UTF-8
Content-Length: 758
```

```
<?xml version="1.0" encoding="UTF-8"?>
<sms:outboundSMSMessageRequest xmlns:sms="urn:oma:xml:rest:sms:1">
       <address>6513507551001</address>
       <senderAddress>861111</senderAddress>
       <senderName>1234</senderName>
       <outboundSMSTextMessage>
              <message>TESTTTTTTTTTT......</message>
       </outboundSMSTextMessage>
</sms:outboundSMSMessageRequest>
```

Table 1-3 Request format

Element	Description
Request line	The request line consists of: • Request method: includes the POST, GET, PUT, and DELETE. • Request URI: address used by the server to provide services. • HTTP protocol version: HTTP/1.1.
Message header	Parameters in the message header includes common HTTP message header parameters and SDP-defined parameters. SDP-defined parameters are used for access authentication and SDP service processing.
Message body	Parameters in the message body comply with the ParlayREST protocol. For details about the parameters, see the request topics.

NOTE

When Apps invoke SDP APIs, requests cannot contain the following XML characters: & > < ' "

If the preceding characters are really required, you must add escape characters before such characters. Otherwise, SDP APIs will fail to be invoked.

1.4 Response Format

A ParlayREST response provided by the SDP consists of the status line, message header, and message body.

Success Response Format

The SDP provides the ParlayREST API success responses in the following format:

```
HTTP/1.1 201 Created
Content-Type: application/xml; charset=UTF-8
Location:
http://127.0.0.1:8312/1/smsmessaging/outbound/861234/requests/60000120040113042710
3027000041
Connection: close
Content-Length: 197

<?xml version="1.0" encoding="utf-8" ?>
<resourceReference>
```

```
<resourceURL>http://127.0.0.1:8312/1/smsmessaging/outbound/861234/requests/6000012
00401130427103027000041</resourceURL>
</resourceReference>
```

Table 1-4 Success response format

Element	Description
Status line	The status line consists of: • HTTP protocol version: HTTP/1.1. • Response status code: complies with the HTTP protocol. • Response status description: description of a status code, which complies with the HTTP protocol.
Message header	Parameters in the message header includes common HTTP message header parameters and SDP-defined parameters. SDP-defined parameters are used for access authentication and SDP service processing.
Message body	Parameters in the message body comply with the ParlayREST protocol. For details about the parameters, see the response topics.

Error Response Format

The SDP provides the Parlay X 2.1 API error responses in the following format:

```
<requestError>
   <errorType>
      <parameter>...</parameter>
...
   </errorType>
</requestError>
```

Table 1-5 Error response format

Element	Description
<requestError>	Root element in an error response, which specifies the error type.
<errorType>	Error type. The options are **serviceException** and **policyException**. This element consists of the <messageId>, <text>, and <variables> elements that are used to define error codes and error details.
<messageId>	Result code. For details about error responses, see4 API Error Responses.
text	Error description.The value can contain the variable %# in definition.
variables	Value of the variable defined in the value of **text**.

1.5 Status Codes

Status codes sent by the SDP comply with the HTTP protocol. Table 1-6 lists the status codes.

Table 1-6 HTTP status codes

Status Code	Description
200	Success.
201	Created.
204	No Content.
304	ConditionNotMet - Not Modified: The condition specified in the conditional header(s) was not met for a read operation.
401	Authentication failure.
403	Application don't have permissions to access resource due to the policy constraints (request rate limit, etc).
404	Not Found - The specified resource does not exist.
405	Method not allowed by the resource.
409	Conflict.
411	Length Required: The Content-Length header was not specified.
412	Precondition Failed: The condition specified in the conditional header(s) was not met for a write operation.
413	RequestBodyTooLarge - Request Entity Too Large: The size of the request body exceeds the maximum size permitted.
416	InvalidRange - Requested Range Not Satisfiable: The range specified is invalid for the current size of the resource.500 - Internal server error.
503	ServerBusy - Service Unavailable: The server is currently unable to receive requests. Please retry your request.

2 APIs for Receiving SMS Messages

2.1 Process

Notify Mode

The process of the App receiving SMS messages in Notify mode consists of the following main steps:

- Subscribing to MO SMS message notification: After the subscription, the SDP notifies the App of MO SMS messages immediately when receiving them from users.
- Receiving MO SMS messages: The App receives MO SMS messages from the SDP in real time.
- Unsubscribing from MO SMS message notification: After the unsubscription, the SDP no longer notifies the App of MO SMS messages.

Figure 2-1 shows the process of receiving SMS messages in Notify mode.

Figure 2-1 Process of receiving SMS messages in Notify mode

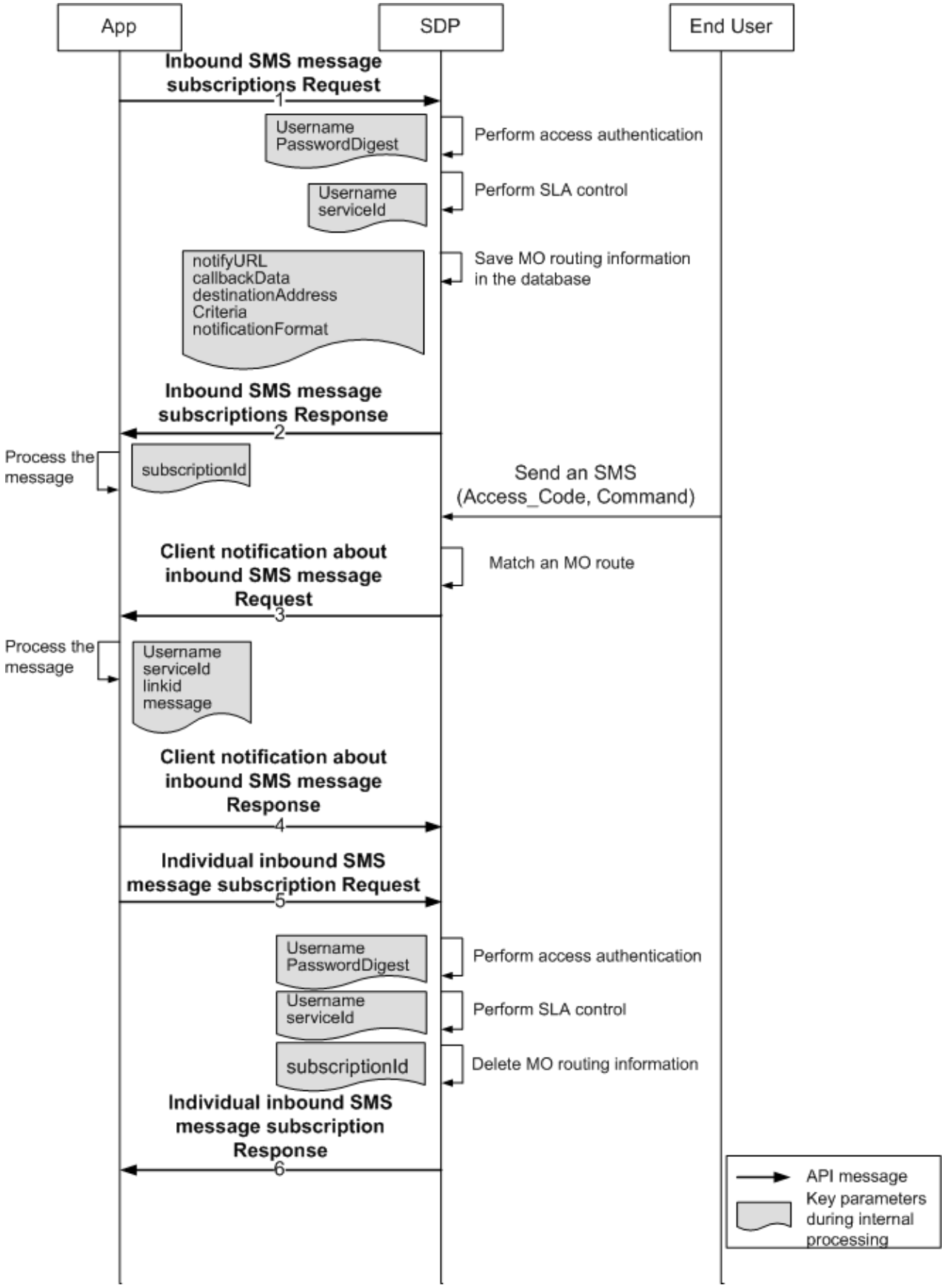

Table 2-1 describes the process.

Table 2-1 Description for the process of receiving SMS messages in Notify mode

Step	Description
1-2	• The App sends a request to the SDP to subscribe to MO SMS message notification. • The SDP performs authentication and service level agreement (SLA) control based on fields in the request, saves MO routing information in the database, and sends a response to the App.
3-4	• The SDP receives an MO SMS message from a user, matches an MO route based on the access code and command word, and sends a notification of the SMS message to the App. • The App parses the notification and sends a response to the SDP.
5-6	• The App sends a request to the SDP to unsubscribe from MO SMS message notification when the App is to be brought offline. • The SDP performs authentication and SLA control based on fields in the request, deletes MO routing information from the database, and sends a response to the App.

Get Mode

Figure 2-2 shows the process of the App receiving SMS messages in Get mode.

Figure 2-2 Process of receiving SMS messages in Get mode

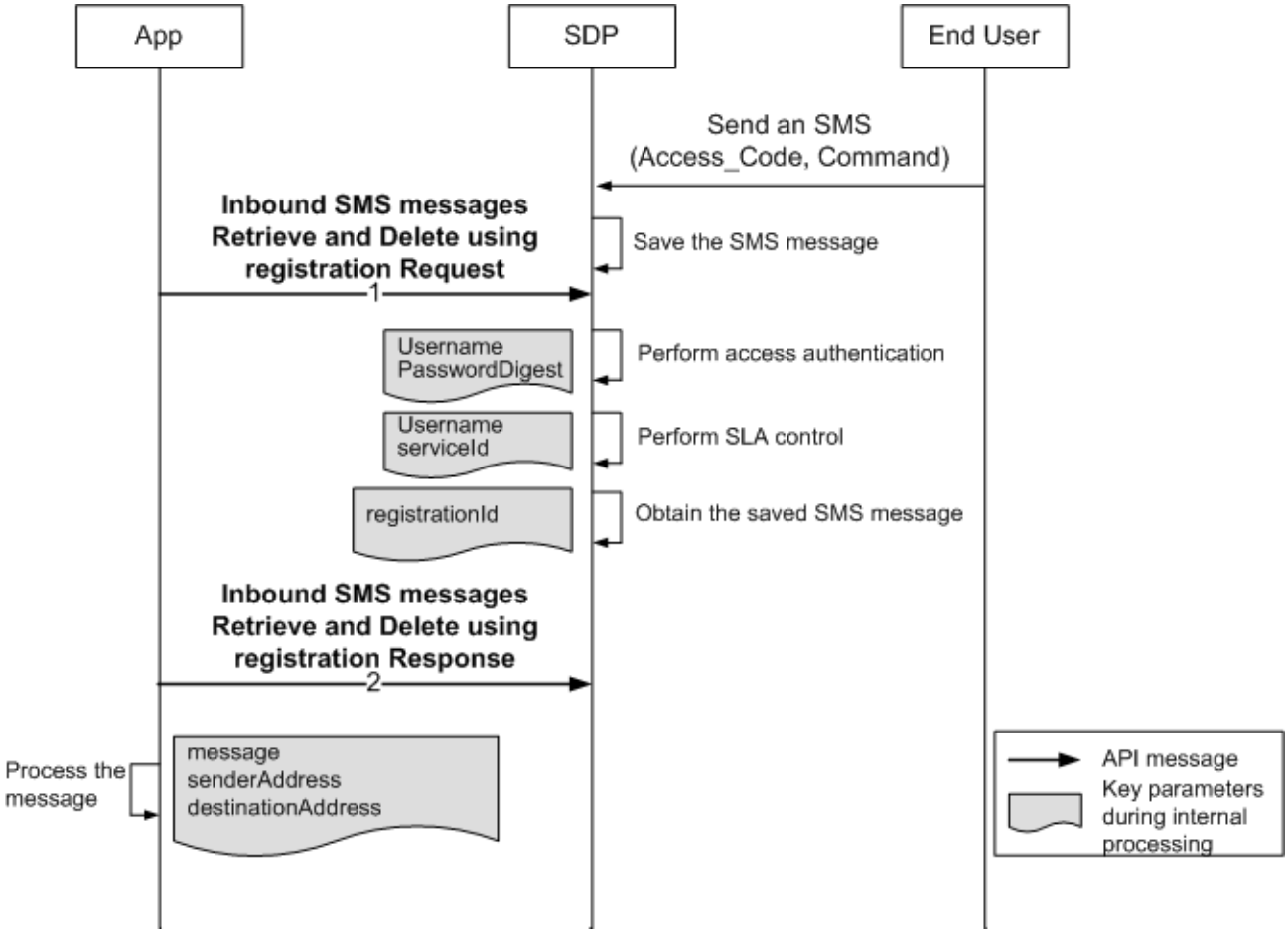

Table 2-2 describes the process.

Table 2-2 Description for the process of receiving SMS messages in Get mode

Step	Description
1-2	• The SDP receives MO SMS messages from users and saves the messages for a specific period (48 hours by default). • The App sends a request to the SDP at scheduled time to obtain SMS messages. • The SDP performs authentication and SLA control based on fields in the request, obtains the saved SMS messages, and sends a response to the App. • The App processes the response.

2.2 Inbound SMS message subscriptions

2.2.1 Function

The App (functioning as the client) invokes the Inbound SMS message subscriptions API to subscribe to MO SMS message notification on the SDP (functioning as the server).

The Inbound SMS message subscriptions API sends the routing information for the App to receive MO SMS message notifications to the SDP. When receiving the API request, the SDP saves the MO routing information of the App. After MO SMS message notification is subscribed, the SDP sends MO SMS messages received from users to the App based on the MO routing information.

Partners must code the App based on the API field requirements so that the App can send correct requests to the SDP. The SDP send a response within 60 seconds by default.

2.2.2 Request Method

Set it to **POST**.

2.2.3 Request URI

The request URI is the destination URI of Inbound SMS message subscriptions requests sent by the App to the SDP. The URI is provided by the SDP in the following format:

http://*IP*:*Port*/*apiVersion*/smsmessaging/inbound/subscriptions

In the format, **IP** and **Port** indicate the service IP address and ParlayREST port number of the API provided by the SDP (Contact carriers to obtain the IP address and port number) and **apiVersion** has a fixed value of **1**.

2.2.4 Request

The App functions as the client and sends a Inbound SMS message subscriptions request to the SDP to subscribe to MO SMS message notification.

Example

Request examples in different request body encoding formats are as follows:

- In XML encoding format:

```
POST /1/smsmessaging/inbound/subscriptions HTTP/1.1
Authorization: WSSE realm="SDP",profile="UsernameToken"
X-WSSE: UsernameToken
Username="35000001",PasswordDigest="D6ri9fAmdFhGijaU6c+9myLB/9s=",Nonce="201304
2718472200001", Created="2013-04-27T18:47:22Z"
X-RequestHeader: request ServiceId="35000001000001"
Accept-Encoding: gzip,deflate
Accept: application/xml
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.137.213.125:14312
Content-Type: application/xml; charset=UTF-8
Content-Length: 389
<sms:subscription xmlns:sms="urn:oma:xml:rest:sms:1">
       <callbackReference>
```

```
                <notifyURL>http://10.135.178.84:9088/</notifyURL>
                <callbackData>123</callbackData>
                <notificationFormat>xml</notificationFormat>
        </callbackReference>
        <destinationAddress>1234</destinationAddress>
        <criteria>order</criteria>
</sms:subscription>
```

- In JSON encoding format:

```
POST /1/smsmessaging/inbound/subscriptions HTTP/1.1
Authorization: WSSE realm="SDP",profile="UsernameToken"
X-WSSE: UsernameToken
Username="35000001",PasswordDigest="B6RoL0jsxHTCSL2nXb9911FOe80=",Nonce="201304
2718511800001", Created="2013-04-27T18:51:18Z"
X-RequestHeader: request ServiceId="35000001000001"
Accept-Encoding: gzip,deflate
Accept: application/json
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.137.213.125:14312
Content-Type: application/json; charset=UTF-8
Content-Length: 315
{"subscription": {
   "callbackReference": {
       "notifyURL": "http://10.135.178.84:9088/",
       "callbackData":"123",
       "notificationFormat":"json"
                    },
   "destinationAddress":[ "1111"],
   "criteria": "order"
              }
}
```

- In URL encoding format:

```
POST /1/smsmessaging/inbound/subscriptions HTTP/1.1
Authorization: WSSE realm="SDP",profile="UsernameToken"
X-WSSE: UsernameToken
Username="35000001",PasswordDigest="B6RoL0jsxHTCSL2nXb9911FOe80=",Nonce="201304
2718511800001", Created="2013-04-27T18:51:18Z"
X-RequestHeader: request ServiceId="35000001000001"
Accept-Encoding: gzip,deflate
Accept: application/x-www-form-urlencoded
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.137.213.125:14312
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Content-Length: 138
destinationAddress=1111&criteria=1111&notifyURL=http://10.135.165.4:8080/Delive
ryInfoNotification&callbackData=1234&notificationFormat=XML
```

Message Header Parameters

Table 2-3 describes parameters in the message header.

Table 2-3 Parameters in the message header

Header Field	Parameter	Type	Length	Level of Requirement	Description
Accept	N/A	String	16	Optional	Format in which responses are encoded. The options are as follows: • application/xml • application/json When this parameter is not transmitted, the SDP uses the **application/xml** mode to encode responses by default. [Example] application/xml
Content-Type	N/A	String	33	Mandatory	Format in which requests are encoded. • application/xml • application/json • application/x-www-form-urlencoded [Example] application/xml
Content-length	N/A	int	2	Mandatory	Number of bytes in the request body. [Example] 1024
Authorization	/	String	10	Mandatory	Authentication mode. The SDP uses the WSSE authentication mode, in which the **realm** and **profile** parameters are involved. **NOTE** The **Authorization** value indicates the policy used by the SDP to perform authentication at the application layer. The SDP authenticates a third-party in WSSE UsernameToken mode. [Example] WSSE
	realm	String	20	Mandatory	Party that performs authentication. Set it to **SDP**. [Example] SDP
	profile	String	20	Mandatory	Third-party access mechanism. Set it to **UsernameToken**. [Example] UsernameToken

Header Field	Parameter	Type	Length	Level of Requirement	Description
X-WSSE	/	String	30	Mandatory	WSSE authentication flag. To enable the SDP to perform authentication at the application layer in the WSSE UsernameToken mode, set this parameter to **UsernameToken**. [Example] UsernameToken
	Username	String	21	Mandatory	Partner ID. The ID is automatically allocated by the SDP to partners after successful registration. To obtain the ID: • An SP can log in to the SDP management portal and query account information, or log in to the mailbox used for registration and view the email notification received after successful registration. • A Developer can log in to the Developer Portal and query account information, or log in to the mailbox used for registration and view the email notification received after successful registration. • An Enterprise must contact the carrier. [Example] 000201
	PasswordDigest	String	100	Mandatory	Authentication key for the SDP to authenticate partners. The value is a character string encrypted from **Nonce** + **Created** + **Password** by SHA-1 and Base64 in sequence. The encryption formula is as follows: PasswordDigest = Base64(SHA-1(Nonce + Created + Password)). In the formula: • **Nonce** and **Created**: random number and its generation time. • **Password**: password for partners to access the SDP. – An SP can obtain the password from the email notification received after successful registration. – A Developer can log in to the Developer Portal, choose **Member** Center > **Account** > **Registration Information** > **Invoke Password**, and set the password.

Header Field	Parameter	Type	Length	Level of Requirement	Description
					– An Enterprise must contact the carrier to obtain the password. [Example] e6434ef249df55c7a21a0b45758a39bb
	Nonce	String	30	Mandatory	Random number generated by an App when sending a message. [Example] 66C92B11FF8A425FB8D4CCFE0ED9ED1F
	Created	String	20	Mandatory	UTC time when the **Nonce** value is generated. [Format] yyyy-MM-dd'T'HH:mm:ss'Z' [Example] 2011-03-22T13:47:56Z
X-RequestHeader	/	String	10	Mandatory	Type of the message sent by a third party. Set it to **request**. [Example] request
	ServiceId	String	21	Conditional	Service ID. The ID is automatically allocated by the SDP to services after successful release. An SP can log in to the SDP management portal and query service information for the ID. This parameter is mandatory in a request sent by an SP and can be left empty in a request sent by a Developer or an Enterprise. [Example] 35000001000001

Message Body Parameters

Table 2-4 describes the parameter structure of the resourceReference type in the message body.

Table 2-4 Parameter structure of the resourceReference type

Parameter	Type	Length	Level of Requirement	Description
callbackReference	common:CallbackReference	N/A	Mandatory	Reference information. Reference information contains the App service address, API name, and correlator ID that a partner provides for the SDP. Based on the reference information, the SDP sends an MO SMS message notification to the matching App when receiving an SMS message from a user. The **callbackReference** parameter is of the CallbackReference type and contains multiple sub-parameters. For details about the CallbackReference type, see Table 2-5.
destinationAddress	xsd:anyURI [1..unbounded]	20	Mandatory	Access code. The value is planned and allocated by carriers. The SDP uses the access code and command word to match an MO route and routes user requests to the App. To obtain the access code: • An SP can log in to the SDP management portal and query service information. SPs can extend access codes allocated by carriers. In an extended access code, the prefix is allocated by carriers and the extension is defined by SPs. • A Developer or an Enterprise must contact the carrier. [Example] 1234501
criteria	xsd:string	50	Optional	Service ordering or subscription command word. Users send SMS messages containing command words to order or subscribe to services. The SDP uses the access code and command word to match an MO route and routes user requests to the App. The command word is defined by SPs during service release. An SP can log in to the SDP management portal and query service information for the command word. This parameter is mandatory in a request sent by an SP who has configured the command word during service release. This parameter can be left empty in a request sent by a Developer, an Enterprise, or an SP who does

Parameter	Type	Length	Level of Requirement	Description
				not configure the command word during service release. [Example] demand

NOTE
A root node named **subscription** must be defined for the subscription type.

Table 2-5 describes the parameter structure of the CallbackReference type.

Table 2-5 Parameter structure of the CallbackReference type

Parameter	Type	Length	Level of Requirement	Description
notifyURL	xsd:anyURI	512	Mandatory	Service address to which an SMS message is sent. [Example] http://10.138.38.139:9080/notify
callbackData	xsd:string	64	Optional	MO SMS message notification correlator ID. When sending an MO SMS message notification to an App, the SDP also sends this parameter value to the App. [Example] 123456
notificationFormat	common:NotificationFormat	N/A	Optional	Encoding format of the body of an MO SMS message sent by the SDP to App. [Enumerated values of **NotificationFormat**] • XML • JSON [Example] XML

2.2.5 Response

The SDP functions as the server, processes Inbound SMS message subscriptions Request messages received from the App, and sends Inbound SMS message subscriptions Response messages to the App.

This topic provides a success response example. If a request fails, the SDP sends an error response that contains an error code. For details about error responses, see API Error Responses.

Example

Response examples in different response body encoding formats are as follows:

- In XML encoding format:

```
HTTP/1.1 201 Created
Content-Type: application/xml; charset=UTF-8
Connection: close
Content-Length: 155
<resourceReference>

<resourceURL>http://127.0.0.1:8312/1/messaging/inbound/subscriptions/1000130606
2026570000103502</resourceURL>
</resourceReference>
```

- In JSON encoding format:

```
HTTP/1.1 201 Created
Content-Type: application/json; charset=UTF-8
Connection: close
Content-Length: 139
{
       "resourceReference": {
              "resourceURL":
"http://127.0.0.1:8312/1/smsmessaging/inbound/subscriptions/6000130427110154000
0776500"
       }
}
```

Message header Parameters

Table 2-6 describes parameters in the message header.

Table 2-6 Parameters in the message header

Parameter	Type	Level of Requirement	Description
Content-Type	xsd:string	Mandatory	Format in which the response is encoded. The options are as follows: • application/xml • application/json [Example] application/xml

Parameter	Type	Level of Requirement	Description
Content-length	int	Mandatory	Number of bytes in the response body. [Example] 1024

Message Body Parameters

Table 2-7 describes the parameter structure of the resourceReference type in the message body.

Table 2-7 Parameter structure of the resourceReference type

Parameter	Type	Length	Level of Requirement	Description
resourceURL	xsd:string	512	Mandatory	Request URL for the App to unsubscribe from the MO SMS message notification function. [Format] http://*serverRoot*/*apiVersion*/smsmessaging/inbound/subscriptions/*subscriptionId* In the format: • *serverRoot*: root address for the SDP to provide services, which contains the IP address and port number. • *apiVersion*: API version number. The fixed value is **1**. • *subscriptionId*: unique ID of a subscription request. When the App requests unsubscription, the unsubscription request must contains this parameter. [Example] http://10.137.213.41:40312/1/smsmessaging/inbound/subscriptions/1000110919070646000001740 0

NOTE
A root node named **resourceReference** must be defined for the ResourceReference type.

2.2.6 Error Codes

Table 2-8 describes the Inbound SMS message subscriptions API error codes that the SDP may return upon an exception. For details about the error codes, see the *SDP Solution Error Code Reference*.

Table 2-8 Inbound SMS message subscriptions error codes

Error Code	Description
SVC0001	Saving SMS notification fail, the record is %1.
	Duplicate SMS notification, criteria is %1, centralCode is %2.
SVC0002	CallbackReference is null.
	CallbackData of CallbackReference is invalid, the value is %1.
	NotifyUrl is null.
	NotifyUrl %1 is invalid.
	NotifyUrl of CallbackReference is too long, the value is %1.
	NotificationFormat %1 is invalid.
SVC0901	UserName is null!
	UserName %1 is invalid!
	Service ID is null!
	Service ID %1 is invalid!
SVC0905	Authentication Failed, cause by SP,because of timestamp is not valid.

2.3 Client notification about inbound SMS message

2.3.1 Function

The SDP (functioning as the client) invokes the Client notification about inbound SMS message API to send MO SMS messages to the App (functioning as the server).

After the App subscribes to MO SMS message notification through the 2.2 Inbound SMS message subscriptions API, the SDP invokes the Client notification about inbound SMS message API to send MO SMS messages received from users to the App. The App must send a response to the SDP within 30 seconds.If the MO SMS messages fail to be sent, the SDP resends the messages to the App when any of the cached message resending criteria is met. Cached SMS messages can be resent for a maximum of five times. SMS messages can be resent at least 1800 seconds after a sending failure.

Partners must code the App based on the API field requirements so that the App can correctly parse and respond to requests received from the SDP.

2.3.2 Request Method

Set it to **POST**.

2.3.3 Request URI

The request URI is the destination URI of Client notification about inbound SMS message request messages sent by the SDP to the App. The URI is defined by the App.

2.3.4 Request

The SDP functions as the client and sends a Client notification about inbound SMS message request to the App to report an MO SMS message.

Example

Request examples in different request body encoding formats are as follows:

- In XML encoding format:

```
POST / HTTP/1.1
Authorization: WSSE realm="SDP",profile="UsernameToken"
X-WSSE: UsernameToken Username="35000001",
PasswordDigest="u6b6k1L3oBazbVPHfb2fJpQVbOQ=", Nonce="2013042718341400001",
Created="2013-04-27T18:34:14Z"
X-NotifyHeader: notify ServiceId="00020100000001" LinkId="12132131"
TransId="10000120010111062302172100011"
Content-Type: application/xml; charset=UTF-8
Host: 10.135.178.84:9088
Connection: close
Content-Length: 377
<?xml version="1.0" encoding="utf-8" ?>
<ns1:inboundSMSMessageNotification xmlns:ns1="urn:oma:xml:rest:sms:1">
  <callbackData>123</callbackData>
  <inboundSMSMessage>
    <destinationAddress>1111</destinationAddress>
    <senderAddress>6513807551001</senderAddress>
    <message>test</message>
    <messageId>60000120040113042710541700012l</messageId>
  </inboundSMSMessage>
</ns1:inboundSMSMessageNotification>
```

- In JSON encoding format:

```
POST / HTTP/1.1
Authorization: WSSE realm="SDP",profile="UsernameToken"
X-WSSE: UsernameToken Username="35000001",
PasswordDigest="u6b6k1L3oBazbVPHfb2fJpQVbOQ=", Nonce="2013042718341400001",
Created="2013-04-27T18:34:14Z"
X-NotifyHeader: notify ServiceId="00020100000001" LinkId="12132131"
TransId="10000120010111062302172100011"
Content-Type: application/json; charset=UTF-8
Host: 10.135.178.84:9088
Connection: close
Content-Length: 253
{
       "inboundSMSMessageNotification": {
              "callbackData": "123",
              "inboundSMSMessage": {
                     "destinationAddress": "1111",
                     "senderAddress": "6513807551001",
```

```
                    "message": "ABC DE ABC",
                    "messageId": "600001200301130427104756000111"
                }
            }
        }
```

Message Header Parameters

Table 2-9 describes parameters in the message header.

Table 2-9 Parameters in the message header

Header Field	Parameter	Type	Length	Level of Requirement	Description
Content-Type	N/A	String	16	Mandatory	Format in which responses are encoded. The options are as follows: • application/xml • application/json [Example] application/xml
Content-length	N/A	int	2	Mandatory	Number of bytes in the request body. [Example] 1024
Authorization	/	String	10	Mandatory	Authentication mode. The SDP uses the WSSE authentication mode, in which the **realm** and **profile** parameters are involved. **NOTE** The **Authorization** value indicates the policy used by the SDP to perform authentication at the application layer. The SDP authenticates a third-party in WSSE UsernameToken mode. [Example] WSSE
	realm	String	20	Mandatory	Set it to **SDP**. [Example] SDP
	profile	String	20	Mandatory	SDP access mechanism. Set it to **UsernameToken**. [Example] UsernameToken
X-WSSE	/	String	30	Mandatory	WSSE authentication flag. To enable the SDP to perform authentication at

Header Field	Parameter	Type	Length	Level of Requirement	Description
					the application layer in the WSSE UsernameToken mode, set this parameter to **UsernameToken**. [Example] UsernameToken
	Username	String	21	Mandatory	Reverse authentication ID for the App to authenticate the SDP. The ID is set by SPs during registration. An SP can log in to the SDP management portal and query account information for the ID. This parameter is mandatory in a request sent to an SP who has configured authentication information during registration. This parameter can be left empty in a request sent to a Developer, an Enterprise, or an SP who does not configure authentication information. [Example] sdp
	PasswordDigest	String	100	Mandatory	Reverse authentication key for the App to authenticate the SDP. The value is a character string encrypted from **Nonce** + **Created** + **Password** by SHA-1 and Base64 in sequence. The encryption formula is as follows: PasswordDigest = Base64(SHA-1(Nonce + Created + Password)). In the formula: • **Nonce** and **Created**: random number and its generation time. • **Password**: access password allocated by an SP to the SDP. An SP can obtain the password from the email notification received after successful registration. This parameter is mandatory in a request sent to an SP who has configured authentication information during registration. This parameter can be left empty in a request sent to a Developer, an Enterprise, or an SP who does not configure authentication information. [Example] 206D88BB7F3D154B130DD6E1E0B8828B
	Nonce	String	30	Mandatory	Random number generated by SDP when sending a message. [Example]

Header Field	Parameter	Type	Length	Level of Requirement	Description
					66C92B11FF8A425FB8D4CCFE0ED9ED1F
	Created	String	20	Mandatory	UTC time when the **Nonce** value is generated. [Format] yyyy-MM-dd'T'HH:mm:ss'Z' [Example] 2011-03-22T13:47:56Z
X-NotifyHeader	/	String	10	Mandatory	Type of the message sent by the SDP. Set it to **notify**. [Example] notify
	TransId	String	30	Mandatory	Transaction ID. The ID is automatically generated by the SDP and is used only to trace messages during the SDP commissioning. The App ignores this parameter. [Example] 10000120010111062302172100011
	ServiceId	String	21	Conditional	Service ID. The ID is automatically allocated by the SDP to services after successful release. An SP can log in to the SDP management portal and query service information for the ID. This parameter is mandatory in a request sent by an SP and can be left empty in a request sent by a Developer or an Enterprise. [Example] 35000001000001
	LinkId	String	20	Conditional	Service order ID. The ID is automatically generated by the SDP when a user orders a service in the SDP. This parameter is mandatory during service ordering by SMS message. [Example] 12345678901111

Message Body Parameters

Table 2-10 describes the parameter structure of the InboundSMSMessageNotification type in the message body.

Table 2-10 Parameter structure of the InboundSMSMessageNotification type

Parameter	Type	Length	Level of Requirement	Description
callbackData	xsd:string	64	Optional	MO SMS message correlator ID. This parameter is set by an SP when the SP sends a request to subscribe to MO SMS message notification. The SDP sends this parameter value when sending an MO SMS message notification. [Example] 12345
inboundSMSMessage	InboundSMSMessage	N/A	Mandatory	SMS message information. This parameter contains the SMS message content, sender's mobile number, access code, and time information. The **inboundSMSMessage** parameter is of the inboundSMSMessage type and contains multiple sub-parameters. For details about the inboundSMSMessage type, see Table 2-11.

NOTE
A root node named **inboundSMSMessageNotification** must be defined for the InboundSMSMessageNotification data structure.

Table 2-11 describes the parameter structure of the InboundSMSMessage type.

Table 2-11 Parameter structure of the InboundSMSMessage type

Parameter	Type	Length	Level of Requirement	Description
destinationAddress	xsd:anyURI	20	Mandatory	Access code. The value is planned and allocated by carriers. The SDP uses the access code and command word to match an MO route and routes user requests to the App. To obtain the access code: • An SP can log in to the SDP management portal and query service information. SPs can extend access codes allocated by carriers. In an extended access code, the prefix is allocated by carriers and the extension

Parameter	Type	Length	Level of Requirement	Description
				is defined by SPs. • A Developer or an Enterprise must contact the carrier. [Format] tel:*Access code* [Example] tel:1234501
senderAddress	xsd:anyURI	30	Mandatory	Mobile number of the sender. [Format] tel:*[Prefix][Country code][Mobile number]* In the format, *[Prefix]* and *[Country code]* are optional. The value of *[Prefix]*, if contained, can be **+**, **+0**, **+00**, **0**, or **00**. [Example] tel:8612312345678
Message	xsd:string	700	Mandatory	SMS message content. [Format] *[Command word] [Message body]* In the format, *[Command word]* is optional. Its value can be a service ordering or subscription command word in a request sent to an SP, and is left empty in a request sent to a Developer or an Enterprise. [Example] order Hello world In the example, **order** is a service ordering command word.
dateTime	xsd:dateTime	30	Optional	Date and time (UTC time) when the SDP receives the SMS message. [Format] *yyyy-MM-dd*T*HH:mm:ss.SSSZ* [Example] 2010-08-09T00:00:00.000+08:00
messageId	xsd:string	30	Optional	Unique ID generated by the SDP for an MO SMS message. [Example] 60000120030113042710475600011 1

2.3.5 Response

The App functions as the server, processes the request messages received from the SDP, and sends the response messages to the SDP.

The response is constructed based on the SDP specification by the partner that provides the App.

Example

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

2.3.6 Error Codes

The App returns error codes to the SDP when an exception occurs in response to Client notification about inbound SMS message requests. The error codes are defined by partners.

2.4 Individual inbound SMS message subscription

2.4.1 Function

The App (functioning as the client) invokes the Individual inbound SMS message subscription API to unsubscribe from MO SMS message notification on the SDP (functioning as the server). This API is invoked by the App when it is to be brought offline.

After MO SMS message notification is unsubscribed, the SDP does not send MO SMS messages received from users to the App.

Partners must code the App based on the API field requirements so that the App can send correct requests to the SDP. The SDP send a response within 60 seconds by default.

2.4.2 Request Method

Set it to **DELETE**.

2.4.3 Request URI

The request URI is the destination URI of Individual inbound SMS message subscription requests sent by the App to the SDP. The URI is provided by the SDP in the following format:

http://*IP*:*Port*/*apiVersion*/smsmessaging/inbound/subscriptions/*subscriptionId*

In the format, *IP* and *Port* indicate the service IP address and ParlayREST port number of the API provided by the SDP (Contact carriers to obtain the IP address and port number), *apiVersion* has a fixed value of **1**, and *subscriptionId* indicates the unique subscription ID, which must be the same as the value of *subscriptionId* in the corresponding Inbound SMS message subscriptions response sent by the SDP.

2.4.4 Request

The App functions as the client and sends a Individual inbound SMS message subscription request to the SDP to unsubscribe from MO SMS message notification.

Example

```
DELETE /1/smsmessaging/inbound/subscriptions/600013042710475500003646 00 HTTP/1.1
Authorization: WSSE realm="SDP",profile="UsernameToken"
X-WSSE: UsernameToken
Username="35000001",PasswordDigest="RJbLsXJduR+mjxJ4PeJvTuPDmE8=",Nonce="201304271
8371600001", Created="2013-04-27T18:37:16Z"
X-RequestHeader: request ServiceId="35000001000001"
Accept-Encoding: gzip,deflate
Accept: application/json
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.137.213.125:14312
Content-Type: application/json; charset=UTF-8
Content-Length: 0
```

Message Header Parameters

Table 2-12 describes parameters in the message header.

Table 2-12 Parameters in the message header

Header Field	Parameter	Type	Length	Level of Requirement	Description
Accept	N/A	String	16	Optional	Format in which responses are encoded. The options are as follows: • application/xml • application/json When this parameter is not transmitted, the SDP uses the **application/xml** mode to encode responses by default. [Example] application/xml
Content-Type	N/A	String	33	Mandatory	Format in which requests are encoded. • application/xml • application/json • application/x-www-form-urlencoded [Example] application/xml
Content-length	N/A	int	2	Mandatory	Number of bytes in the request body. [Example] 1024
Authorization	/	String	10	Mandatory	Authentication mode. The SDP uses the WSSE authentication mode, in which the **realm** and **profile** parameters are

Header Field	Parameter	Type	Length	Level of Requirement	Description
					involved. **NOTE** The **Authorization** value indicates the policy used by the SDP to perform authentication at the application layer. The SDP authenticates a third-party in WSSE UsernameToken mode. [Example] WSSE
	realm	String	20	Mandatory	Party that performs authentication. Set it to **SDP**. [Example] SDP
	profile	String	20	Mandatory	Third-party access mechanism. Set it to **UsernameToken**. [Example] UsernameToken
X-WSSE	/	String	30	Mandatory	WSSE authentication flag. To enable the SDP to perform authentication at the application layer in the WSSE UsernameToken mode, set this parameter to **UsernameToken**. [Example] UsernameToken
	Username	String	21	Mandatory	Partner ID. The ID is automatically allocated by the SDP to partners after successful registration. To obtain the ID: • An SP can log in to the SDP management portal and query account information, or log in to the mailbox used for registration and view the email notification received after successful registration. • A Developer can log in to the Developer Portal and query account information, or log in to the mailbox used for registration and view the email notification received after successful registration. • An Enterprise must contact the carrier. [Example] 000201
	Password	Strin	100	Mand	Authentication key for the SDP to authenticate

Header Field	Parameter	Type	Length	Level of Requirement	Description
	Digest	g		atory	partners. The value is a character string encrypted from **Nonce** + **Created** + **Password** by SHA-1 and Base64 in sequence. The encryption formula is as follows: PasswordDigest = Base64(SHA-1(Nonce + Created + Password)). In the formula: • **Nonce** and **Created**: random number and its generation time. • **Password**: password for partners to access the SDP. – An SP can obtain the password from the email notification received after successful registration. – A Developer can log in to the Developer Portal, choose **Member Center** > **Account** > **Registration Information** > **Invoke Password**, and set the password. – An Enterprise must contact the carrier to obtain the password. [Example] e6434ef249df55c7a21a0b45758a39bb
	Nonce	String	30	Mandatory	Random number generated by an App when sending a message. [Example] 66C92B11FF8A425FB8D4CCFE0ED9ED1F
	Created	String	20	Mandatory	UTC time when the **Nonce** value is generated. [Format] yyyy-MM-dd'T'HH:mm:ss'Z' [Example] 2011-03-22T13:47:56Z
X-RequestHeader	/	String	10	Mandatory	Type of the message sent by a third party. Set it to **request**. [Example] request
	ServiceId	String	21	Conditional	Service ID. The ID is automatically allocated by the SDP to services after successful release. An SP can log in to the SDP management portal and query service information for the ID. This parameter is mandatory in a request sent by

Header Field	Parameter	Type	Length	Level of Requirement	Description
					an SP and can be left empty in a request sent by a Developer or an Enterprise. [Example] 35000001000001

Message Body Parameters

None

2.4.5 Response

The SDP functions as the server, processes the Individual inbound SMS message subscription requests received from the App, and sends the responses to the App.

This topic provides a success response example. If a request fails, the SDP sends an error response that contains an error code. For details about error responses, see API Error Responses.

Example

```
HTTP/1.1 200 OK
Connection: close
Content-Length: 0
```

2.4.6 Error Codes

Table 2-13 describes Individual inbound SMS message subscription error codes that the SDP may return upon an exception. For details about the error codes, see the *SDP Solution Error Code Reference*.

Table 2-13 Individual inbound message subscription error codes

Error Code	Description
SVC0002	Correlator is null.
	Correlator %1 has a invalid format.
SVC0901	UserName is null!
	UserName %1 is invalid!
	Service ID is null!
	Service ID %1 is invalid!
	Authentication Failed, cause by SP,because of timestamp expired.

Error Code	Description
	SP ip is null!
	Created is empty in X-WSSE httpheader.
	Nonce is empty in X-WSSE httpheader.
	Sp ip %1 is not accepted!
SVC0905	Authentication Failed, cause by SP,because of timestamp is not valid.

2.5 Inbound SMS messages Retrieve and Delete using registration

2.5.1 Function

The App (functioning as the client) invokes the Inbound SMS messages Retrieve and Delete using registration API to obtain MO SMS messages from the SDP (functioning as the server). The SDP saves MO SMS messages received from the SMSC for only 48 hours.

When the App invokes the Inbound SMS messages Retrieve and Delete using registration API, the SDP sends MO SMS messages to the App. The number of MO SMS messages sent by the SDP to the App is determined by the SLA of the App.

Partners must code the App based on the API field requirements so that the App can send correct requests to the SDP. The SDP send a response within 60 seconds by default.

2.5.2 Request Method

Set it to **POST**.

2.5.3 Request URI

The request URI is the destination URI of Inbound SMS messages Retrieve and Delete using registration requests sent by the App to the SDP. The URI is provided by the SDP in the following format:

http://*IP*:*Port*/*apiVersion*/inbound/registrations/*registrationId*/retrieveAndDeleteMessages

In the format, **IP** and **Port** indicate the service IP address and ParlayREST port number of the API provided by the SDP. *registrationId* indicate the access code planned and allocated by carriers.Contact carriers to obtain the IP address, port number and the access code.

2.5.4 Request

The App functions as the client and sends a Inbound SMS messages Retrieve and Delete using registration request to the SDP to receive MO SMS messages.

Example

Request examples in different request body encoding formats are as follows:

- In XML encoding format:

```
POST /1/smsmessaging/inbound/registrations/861111/retrieveAndDeleteMessages
HTTP/1.1
Authorization: WSSE realm="SDP",profile="UsernameToken"
X-WSSE: UsernameToken
Username="35000001",PasswordDigest="DJ72ERcqL4EM3PzEocFbPkv0834=",Nonce="201304
2718355200001", Created="2013-04-27T18:35:52Z"
X-RequestHeader: request ServiceId="35000001000001"
Accept-Encoding: gzip,deflate
Accept: application/xml
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.137.213.125:14312
Content-Type: application/xml; charset=UTF-8
Content-Length: 252
<?xml version="1.0" encoding="UTF-8"?>
<sms:inboundSMSRetrieveAndDeleteMessageRequest
xmlns:sms="urn:oma:xml:rest:sms:1">
       <retrievalOrder>OldestFirst</retrievalOrder>
       <maxBatchSize>1</maxBatchSize>
</sms:inboundSMSRetrieveAndDeleteMessageRequest>
```

- In JSON encoding format:

```
POST /1/smsmessaging/inbound/registrations/861111/retrieveAndDeleteMessages
HTTP/1.1
Authorization: WSSE realm="SDP",profile="UsernameToken"
X-WSSE: UsernameToken Username="35000001",
PasswordDigest="u6b6k1L3oBazbVPHfb2fJpQVbOQ=", Nonce="2013042718341400001",
Created="2013-04-27T18:34:14Z"
X-RequestHeader: request ServiceId="35000001000001"
Accept-Encoding: gzip,deflate
Accept: application/json
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.137.213.125:14312
Content-Type: application/json; charset=UTF-8
Content-Length: 135
{
   "inboundSMSRetrieveAndDeleteMessageRequest":
   {
      "maxBatchSize":1,
      "retrievalOrder":"OldestFirst"
   }
}
```

- In URL encoding format:

```
POST /1/smsmessaging/inbound/registrations/861111/retrieveAndDeleteMessages
HTTP/1.1
Authorization: WSSE realm="SDP",profile="UsernameToken"
X-WSSE: UsernameToken Username="35000001",
PasswordDigest="u6b6k1L3oBazbVPHfb2fJpQVbOQ=", Nonce="2013042718341400001",
Created="2013-04-27T18:34:14Z"
X-RequestHeader: request ServiceId="35000001000001"
Accept-Encoding: gzip,deflate
```

```
Accept: application/json
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.137.213.125:14312
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Content-Length: 41
retrievalOrder=OldestFirst&maxBatchSize=3
```

Message Header Parameters

Table 2-14 describes parameters in the message header.

Table 2-14 Parameters in the message header

Header Field	Parameter	Type	Length	Level of Requirement	Description
Accept	N/A	String	16	Optional	Format in which responses are encoded. The options are as follows: • application/xml • application/json When this parameter is not transmitted, the SDP uses the **application/xml** mode to encode responses by default. [Example] application/xml
Content-Type	N/A	String	33	Mandatory	Format in which requests are encoded. • application/xml • application/json • application/x-www-form-urlencoded [Example] application/xml
Content-length	N/A	int	2	Mandatory	Number of bytes in the request body. [Example] 1024
Authorization	/	String	10	Mandatory	Authentication mode. The SDP uses the WSSE authentication mode, in which the **realm** and **profile** parameters are involved. **NOTE** The **Authorization** value indicates the policy used by the SDP to perform authentication at the application layer. The SDP authenticates a third-party in WSSE UsernameToken mode. [Example] WSSE

Header Field	Parameter	Type	Length	Level of Requirement	Description
	realm	String	20	Mandatory	Party that performs authentication. Set it to **SDP**. [Example] SDP
	profile	String	20	Mandatory	Third-party access mechanism. Set it to **UsernameToken**. [Example] UsernameToken
X-WSSE	/	String	30	Mandatory	WSSE authentication flag. To enable the SDP to perform authentication at the application layer in the WSSE UsernameToken mode, set this parameter to **UsernameToken**. [Example] UsernameToken
	Username	String	21	Mandatory	Partner ID. The ID is automatically allocated by the SDP to partners after successful registration. To obtain the ID: • An SP can log in to the SDP management portal and query account information, or log in to the mailbox used for registration and view the email notification received after successful registration. • A Developer can log in to the Developer Portal and query account information, or log in to the mailbox used for registration and view the email notification received after successful registration. • An Enterprise must contact the carrier. [Example] 000201
	PasswordDigest	String	100	Mandatory	Authentication key for the SDP to authenticate partners. The value is a character string encrypted from **Nonce** + **Created** + **Password** by SHA-1 and Base64 in sequence. The encryption formula is as follows: PasswordDigest = Base64(SHA-1(Nonce + Created + Password)). In the formula: • **Nonce** and **Created**: random number and its generation time. • **Password**: password for partners to access the

Header Field	Parameter	Type	Length	Level of Requirement	Description
					SDP. –An SP can obtain the password from the email notification received after successful registration. –A Developer can log in to the Developer Portal, choose **Member** Center > **Account** > **Registration Information** > **Invoke Password**, and set the password. –An Enterprise must contact the carrier to obtain the password. [Example] e6434ef249df55c7a21a0b45758a39bb
	Nonce	String	30	Mandatory	Random number generated by an App when sending a message. [Example] 66C92B11FF8A425FB8D4CCFE0ED9ED1F
	Created	String	20	Mandatory	UTC time when the **Nonce** value is generated. [Format] yyyy-MM-dd'T'HH:mm:ss'Z' [Example] 2011-03-22T13:47:56Z
X-RequestHeader	/	String	10	Mandatory	Type of the message sent by a third party. Set it to **request**. [Example] request
	ServiceId	String	21	Conditional	Service ID. The ID is automatically allocated by the SDP to services after successful release. An SP can log in to the SDP management portal and query service information for the ID. This parameter is mandatory in a request sent by an SP and can be left empty in a request sent by a Developer or an Enterprise. [Example] 35000001000001

Message Body Parameters

Table 2-15 describes the parameter structure of the inboundSMSMessageRetrieveAndDeleteRequest type in the message body.

Table 2-15 Parameter structure of the inboundSMSMessageRetrieveAndDeleteRequest type

Parameter	Type	Length	Level of Requirement	Description
retrievalOrder	RetrievalOrder	N/A	Optional	Sequence in which the SDP is required to send SMS messages to the App. The numeration value options are as follows: • **OldestFirst**: sending the first received message first. • **NewestFirst**: sending the last received message first. The SDP does not support this parameter currently. [Example] OldestFirst
maxBatchSize	xsd:int	2	Optional	Maximum number of SMS messages that the SDP is allowed to send to an App. The SDP compares the **maxBatchSize** value with the upper limit specified in the SLA and uses the smaller one as this parameter value. The upper limit specified in the SLA is 10 in the SDP by default. [Example] 20

NOTE
A root node named **inboundSMSMessageRetrieveAndDeleteRequest** must be defined for the inboundSMSMessageRetrieveAndDeleteRequest data structure.

2.5.5 Response

The SDP functions as the server, processes requests received from the App, and sends responses to the App.

This topic provides a success response example and describes parameters in the response. If a request fails, the SDP sends an error response that contains an error code. For details about error responses, see API Error Responses.

Example

Response examples in different response body encoding formats are as follows:

- In XML encoding format:

```
HTTP/1.1 200 OK
Content-Type: application/xml; charset=UTF-8
Connection: close
Content-Length: 668
<?xml version="1.0" encoding="utf-8" ?>
<inboundSMSMessageList>
  <inboundSMSMessage>
    <destinationAddress>1111</destinationAddress>
    <senderAddress>6513807551001</senderAddress>
    <message>ABCDE</message>
    <dateTime></dateTime>

<resourceURL>http://127.0.0.1:8312/1/smsmessaging/inbound/registrations/1111/re
trieveAndDeleteMessages</resourceURL>
    <messageId>20010103130427181100000000003108</messageId>
  </inboundSMSMessage>
  <totalNumberOfPendingMessages>1</totalNumberOfPendingMessages>
  <numberOfMessagesInThisBatch>1</numberOfMessagesInThisBatch>

<resourceURL>http://127.0.0.1:8312/1/smsmessaging/inbound/registrations/861111/
retrieveAndDeleteMessages</resourceURL>
</inboundSMSMessageList>
```

- In JSON encoding format:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Connection: close
Content-Length: 503
{
       "inboundSMSMessageList": {
              "inboundSMSMessages": [{
                     "destinationAddress": "1111",
                     "senderAddress": "6513807551001",
                     "message": "",
                     "dateTime": "",
                     "resourceURL":
"http://127.0.0.1:8312/1/smsmessaging/inbound/registrations/1111/retrieveAndDel
eteMessages",
                     "messageId": "20010101130427181100000000003108"
              }],
              "totalNumber": 1,
              "numberThisBatch": 1,
              "resourceUrl":
"http://127.0.0.1:8312/1/smsmessaging/inbound/registrations/861111/retrieveAndD
eleteMessages"
       }
}
```

Message header Parameters

Table 2-16 describes parameters in the message header.

Table 2-16 Parameters in the message header

Parameter	Type	Level of Requirement	Description
Content-Type	xsd:string	Mandatory	Format in which the response is encoded. The options are as follows: • application/xml • application/json [Example] application/xml
Content-length	int	Mandatory	Number of bytes in the response body. [Example] 1024

Message Body Parameters

Table 2-17 describes the parameter structure of the inboundSMSMessageList type in the message body.

Table 2-17 Parameter structure of the inboundSMSMessageList type

Parameter	Type	Length	Level of Requirement	Description
inboundSMSMessage	InboundSMSMessage[0..unbounded]	N/A	Optional	SMS message content. This parameter contains the SMS message content, sender's mobile number, access code, and time information. The **inboundSMSMessage** parameter is of the InboundSMSMessage type and contains multiple sub-parameters. For details about the InboundSMSMessage type, see Table 2-18.
totalNumberOfPendingMessages	xsd:int	2	Mandatory	Number of SMS messages stored in the SDP when the App sends a request to obtain MO SMS messages. [Example] 1000
numberOfMessagesInThisBatch	xsd:int	2	Mandatory	Number of SMS messages in the response. [Example]

Parameter	Type	Length	Level of Requirement	Description
				10
resourceURL	xsd:String	512	Mandatory	Request URL provided by the SDP to obtain MO SMS messages. [Format] http://*IP*:*Port*/*apiVersion*/inbound/registrations/*registrationId*/retrieveAndDeleteMessages In the format,*registrationId* indicate the access code. [Example] http://127.0.0.1:8312/1/smsmessaging/inbound/registrations/1111/retrieveAndDeleteMessages

NOTE
A root node named **inboundSMSMessageList** must be defined for the InboundSMSMessageList type.

Table 2-18 describes the parameter structure of the InboundSMSMessage type.

Table 2-18 Parameter structure of the InboundSMSMessage type

Parameter	Type	Length	Level of Requirement	Description
destinationAddress	xsd:anyURI	20	Mandatory	Access code. The value is planned and allocated by carriers. The SDP uses the access code and command word to match an MO route and routes user requests to the App. To obtain the access code: • An SP can log in to the SDP management portal and query service information. SPs can extend access codes allocated by carriers. In an extended access code, the prefix is allocated by carriers and the extension is defined by SPs. • A Developer or an Enterprise must contact the carrier. [Format] tel:*Access code* [Example] tel:1234501

Parameter	Type	Length	Level of Requirement	Description
senderAddress	xsd:anyURI	30	Mandatory	Mobile number of the sender. [Format] tel:*[Prefix][Country code][Mobile number]* In the format, *[Prefix]* and *[Country code]* are optional. The value of *[Prefix]* can be **+**, **+0**, **+00**, **0**, or **00**. [Example] tel:8612312345678
message	xsd:string	700	Mandatory	SMS message content. [Example] Hello world
dateTime	xsd:dateTime	30	Optional	Date and time when the SDP receives the SMS message. [Format] *yyyy-MM-dd*T*HH:mm:ss.SSSZ* [Example] 2010-08-09T00:00:00.000+08:00
resourceURL	xsd:anyURI	512	Optional	URL for the App to obtain an MO SMS message from the SDP. [Example] http://127.0.0.1:8312/1/smsmessaging/inbound/registrations/1111/retrieveAndDeleteMessages
messageId	xsd:string	30	Optional	Unique ID generated by the SDP for an SMS message. [Example] 20010101130427181100000000 3108

2.5.6 Error Codes

Table 2-19 describes Inbound SMS messages Retrieve and Delete using registration error codes that the SDP may return upon an exception. For details about the error codes, see the *SDP Solution Error Code Reference*.

Table 2-19 Inbound SMS messages Retrieve and Delete using registration error codes

Error Code	Description
SVC0901	UserName is null!

Error Code	Description
	UserName %1 is invalid!
	Service ID is null!
	Service ID %1 is invalid!
SVC0905	Authentication Failed, cause by SP, because of timestamp is not valid.
	SMS max returned num

3 APIs for Sending SMS Messages

3.1 Process

The App sends SMS messages to users and receives status reports in Notify or Get mode to determine whether the SMS messages are successfully sent.

If the App does not require status reports, the process of sending SMS messages is from steps 1 to 2 in Figure 3-2.

Notify Mode

Figure 3-1 shows the process of the App sending SMS messages and receiving status reports in Notify mode.

Figure 3-1 Process of sending SMS messages and receiving status reports in Notify mode

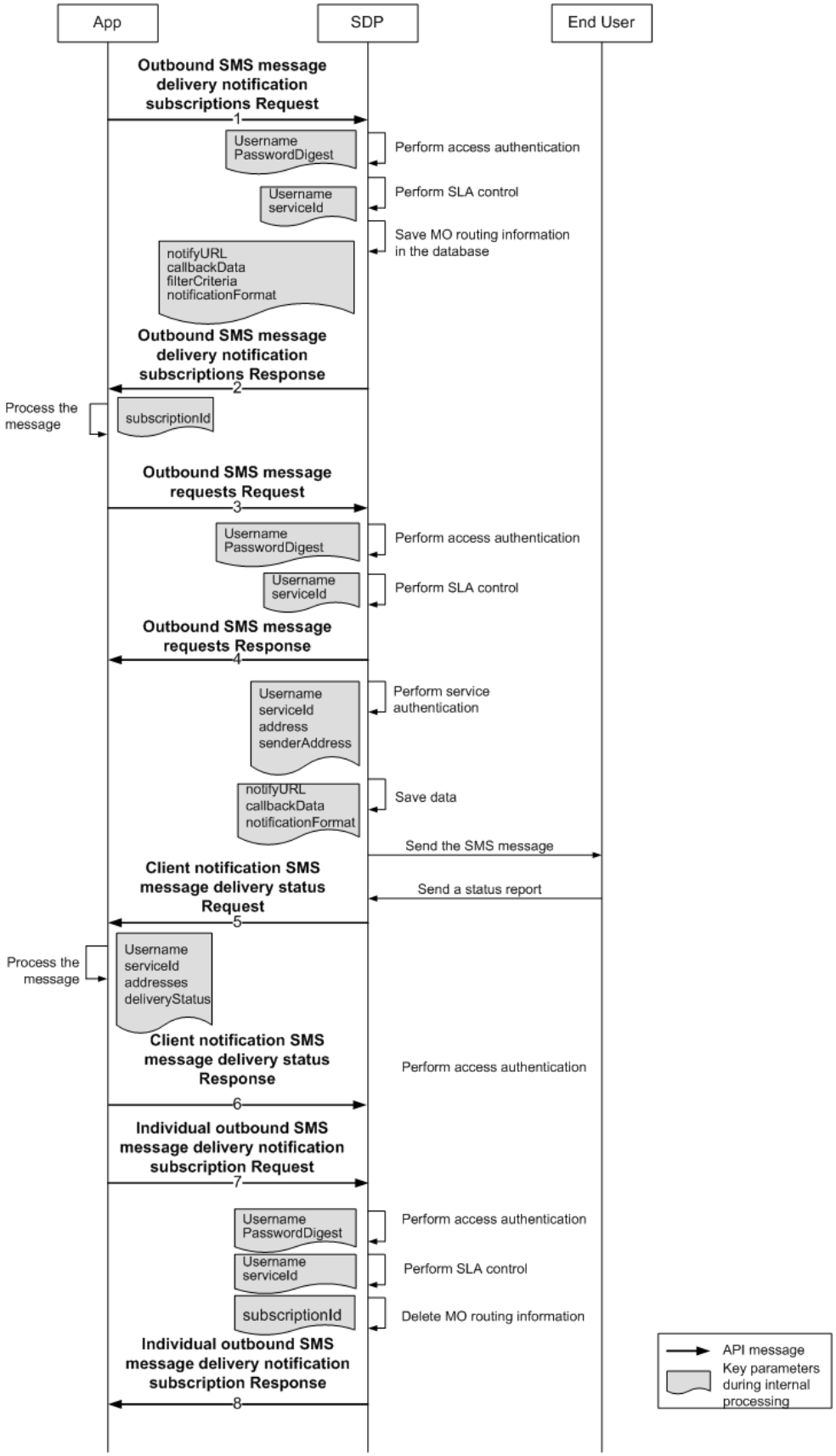

Table 3-1 describes the process.

Table 3-1 Description for the process of sending SMS messages and receiving status reports in Notify mode

Step	Description
1–2	• The App sends a request to the SDP to subscribe to MO status report notification. • The SDP performs authentication and service level agreement (SLA) control based on fields in the request, saves MO routing information in the database, and sends a response to the App. • The App sends a request to the SDP to subscribe to MO SMS message notification.
3-4	• The App sends a request to the SDP to send an SMS message. • The SDP performs authentication and SLA control based on fields in the request and sends a response to the App. Then the SDP performs service authentication, saves the values of **notifyURL** and **callbackData** in the request, and sends the SMS message to the user.
5-6	• The SDP receives a status report and sends a notification of the status report to the App based on **notifyURL**. • The App parses the notification and sends a response to the SDP.
7–8	• The App sends a request to the SDP to unsubscribe from MO status report notification. • The SDP performs authentication and SLA control based on fields in the request, deletes MO routing information from the database, and sends a response to the App.

Get Mode

Figure 3-2 shows the process of the App sending SMS messages and receiving status reports in Get mode.

Figure 3-2 Process of sending SMS messages and receiving status reports in Get mode

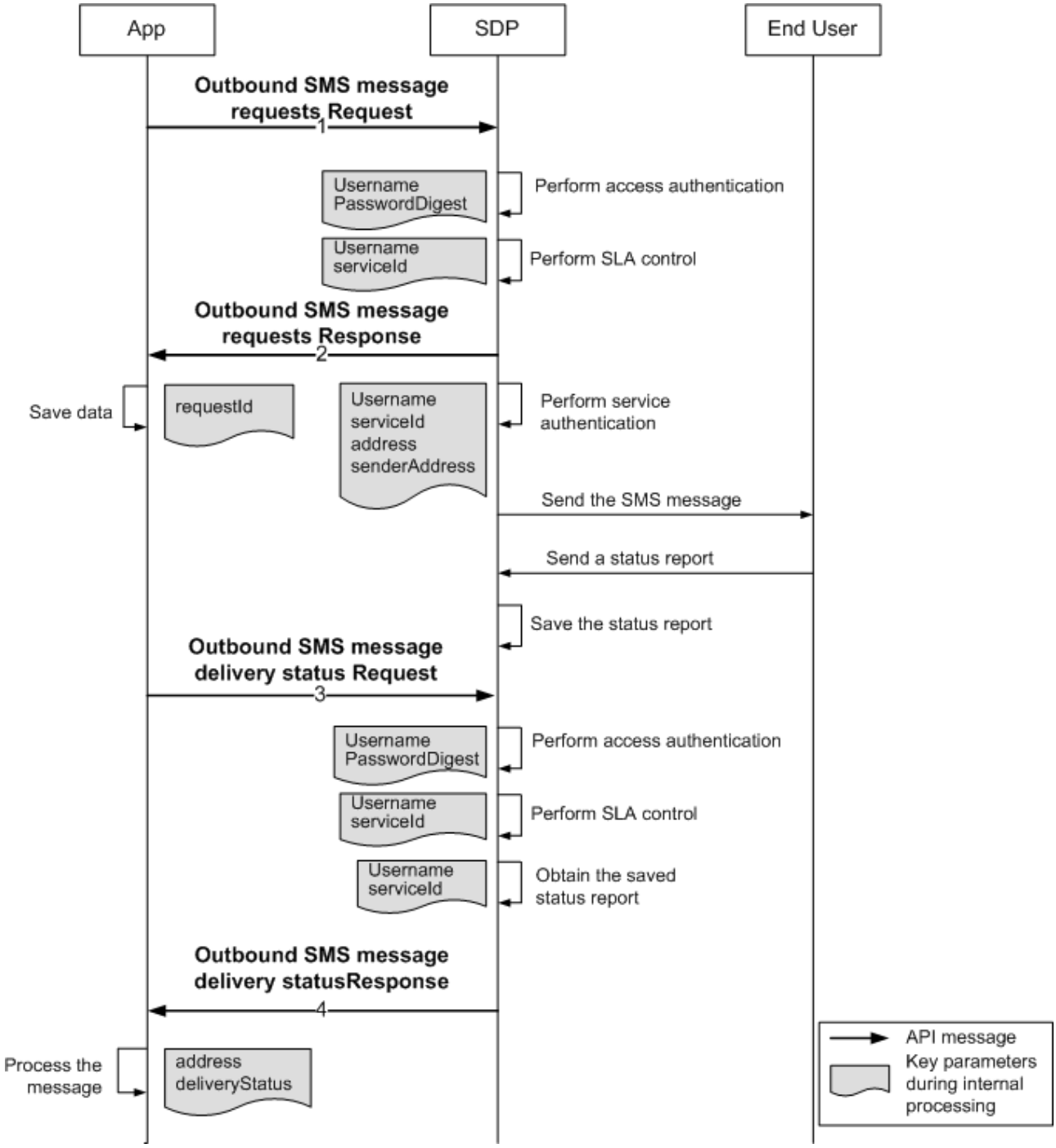

Table 3-2 describes the process.

Table 3-2 Description for the process of sending SMS messages and receiving status reports in Get mode

Step	Description
1-2	• The App sends a request to the SDP to send an SMS message. • The SDP performs authentication and SLA control based on fields in the request and sends a response to the App. Then the SDP performs service authentication and sends the SMS message to the user.
3-4	• The SDP receives a status report from the user and saves the report for a specific period (48 hours by default). • The App sends a request to the SDP at scheduled time to obtain the status report. • The SDP performs authentication and SLA control based on fields in the request, obtains the saved status report, and sends a response to the App. • The App processes the response.

3.2 Outbound SMS message delivery notification subscriptions

3.2.1 Function

The App (functioning as the client) invokes the Outbound SMS message delivery notification subscriptions API to subscribe to MO status report notification on the SDP (functioning as the server).

This API sends the routing information for the App to receive MO status report notifications to the SDP. When receiving a Outbound SMS message delivery notification subscriptions message, the SDP saves the MO routing information of the App. After MO status report notification is enabled, the SDP sends MO status report received from the SMSC to the App based on the MO routing information.

Partners must code the App based on the API field requirements so that the App can send correct requests to the SDP. The SDP send a response within 60 seconds by default.

3.2.2 Request Method

Set it to **POST**.

3.2.3 Request URI

The request URI is the destination URI of Outbound SMS message delivery notification subscriptions requests sent by the App to the SDP. The URI is provided by the SDP in the following format:

http://*IP*:*Port*/*apiVersion*/smsmessaging/outbound/*senderAddress*/subscriptions

In the format, **IP** and **Port** indicate the service IP address and ParlayREST port number of the API provided by the SDP, *senderAddress* indicates the access code planned and provided by

the carrier, and **apiVersion** has a fixed value of **1**. Contact carriers to obtain the IP address, port number and access code.

3.2.4 Request

The App functions as the client and sends a Outbound SMS message delivery notification subscriptions request to the SDP to subscribe to MO SMS messages status report message notification.

Example

Request examples in different request body encoding formats are as follows:

- In XML encoding format:

```
POST /1/smsmessaging/outbound/1234/subscriptions HTTP/1.1
Authorization: WSSE realm="SDP",profile="UsernameToken"
X-WSSE: UsernameToken
Username="35000001",PasswordDigest="Djqi+zzHQW7LFUWSoeW30yn0x3I=",Nonce="201304
2718263400001", Created="2013-04-27T18:26:34Z"
X-RequestHeader: request ServiceId="35000001000001"
Accept-Encoding: gzip,deflate
Accept:
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.135.178.84:9088
Content-Type: application/xml; charset=UTF-8
Content-Length: 380
<sms:deliveryReceiptSubscription xmlns:sms="urn:oma:xml:rest:sms:1">
      <callbackReference>
            <notifyURL>http://10.135.178.84:9088/</notifyURL>
            <callbackData>123</callbackData>
            <notificationFormat>xml</notificationFormat>
      </callbackReference>
      <filterCriteria>6513507551001</filterCriteria>
</sms:deliveryReceiptSubscription>
```

- In JSON encoding format:

```
POST /1/smsmessaging/outbound/861111/subscriptions HTTP/1.1
Authorization: WSSE realm="SDP",profile="UsernameToken"
X-WSSE: UsernameToken Username="35000001",
PasswordDigest="qZlMyqTTNF4HPv7E8nYrAcwZfyc=", Nonce="2013042718064200001",
Created="2013-04-27T18:06:42Z"
X-RequestHeader: request ServiceId="35000001000001"
Accept-Encoding: gzip,deflate
Accept: application/json
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.137.213.125:14312
Content-Type: application/json; charset=UTF-8
Content-Length: 317
{
   "deliveryReceiptSubscription":
   {
      "callbackReference":
      {
         "notifyURL":"http://10.135.178.84:9088/",
         "callbackData":"123",
```

```
            "notificationFormat":"json"
        },
        "filterCriteria":"6513507551001"
    }
}
```

- In URL encoding format:

```
POST /1/smsmessaging/outbound/861111/subscriptions HTTP/1.1
Authorization: WSSE realm="SDP",profile="UsernameToken"
X-WSSE: UsernameToken Username="35000001",
PasswordDigest="qZlMyqTTNF4HPv7E8nYrAcwZfyc=", Nonce="2013042718064200001",
Created="2013-04-27T18:06:42Z"
X-RequestHeader: request ServiceId="35000001000001"
Accept-Encoding: gzip,deflate
Accept: application/json
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.137.213.125:14312
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Content-Length: 142
filterCriteria=1111&notifyURL=http://10.135.165.4:8080/DeliveryInfoNotification
&callbackData=1234&notificationFormat=XML
```

Message Header Parameters

Table 3-3 describes parameters in the message header.

Table 3-3 Parameters in the message header

Header Field	Parameter	Type	Length	Level of Requirement	Description
Accept	N/A	String	16	Optional	Format in which responses are encoded. The options are as follows: • application/xml • application/json When this parameter is not transmitted, the SDP uses the **application/xml** mode to encode responses by default. [Example] application/xml
Content-Type	N/A	String	33	Mandatory	Format in which requests are encoded. • application/xml • application/json • application/x-www-form-urlencoded [Example] application/xml
Content-len	N/A	int	2	Mand	Number of bytes in the request body.

Header Field	Parameter	Type	Length	Level of Requirement	Description
gth				atory	[Example] 1024
Authorization	/	String	10	Mandatory	Authentication mode. The SDP uses the WSSE authentication mode, in which the **realm** and **profile** parameters are involved. **NOTE** The **Authorization** value indicates the policy used by the SDP to perform authentication at the application layer. The SDP authenticates a third-party in WSSE UsernameToken mode. [Example] WSSE
	realm	String	20	Mandatory	Party that performs authentication. Set it to **SDP**. [Example] SDP
	profile	String	20	Mandatory	Third-party access mechanism. Set it to **UsernameToken**. [Example] UsernameToken
X-WSSE	/	String	30	Mandatory	WSSE authentication flag. To enable the SDP to perform authentication at the application layer in the WSSE UsernameToken mode, set this parameter to **UsernameToken**. [Example] UsernameToken
	Username	String	21	Mandatory	Partner ID. The ID is automatically allocated by the SDP to partners after successful registration. To obtain the ID: • An SP can log in to the SDP management portal and query account information, or log in to the mailbox used for registration and view the email notification received after successful registration. • A Developer can log in to the Developer Portal and query account information, or log in to the mailbox used for registration and view the email notification received after successful registration.

Header Field	Parameter	Type	Length	Level of Requirement	Description
					• An Enterprise must contact the carrier. [Example] 000201
	Password Digest	String	100	Mandatory	Authentication key for the SDP to authenticate partners. The value is a character string encrypted from **Nonce** + **Created** + **Password** by SHA-1 and Base64 in sequence. The encryption formula is as follows: PasswordDigest = Base64(SHA-1(Nonce + Created + Password)). In the formula: • **Nonce** and **Created**: random number and its generation time. • **Password**: password for partners to access the SDP. – An SP can obtain the password from the email notification received after successful registration. – A Developer can log in to the Developer Portal, choose **Member Center** > **Account** > **Registration Information** > **Invoke Password**, and set the password. – An Enterprise must contact the carrier to obtain the password. [Example] e6434ef249df55c7a21a0b45758a39bb
	Nonce	String	30	Mandatory	Random number generated by an App when sending a message. [Example] 66C92B11FF8A425FB8D4CCFE0ED9ED1F
	Created	String	20	Mandatory	UTC time when the **Nonce** value is generated. [Format] yyyy-MM-dd'T'HH:mm:ss'Z' [Example] 2011-03-22T13:47:56Z
X-Request Header	/	String	10	Mandatory	Type of the message sent by a third party. Set it to **request**. [Example] request
	Servi	Strin	21	Cond	Service ID.

Header Field	Parameter	Type	Length	Level of Requirement	Description
	ceId	g		itional	The ID is automatically allocated by the SDP to services after successful release. An SP can log in to the SDP management portal and query service information for the ID. This parameter is mandatory in a request sent by an SP and can be left empty in a request sent by a Developer or an Enterprise. [Example] 35000001000001

Message Body Parameters

Table 3-4 describes the parameter structure of the deliveryReceiptSubscription type in the message body.

Table 3-4 Parameter structure of the deliveryReceiptSubscription type

Parameter	Type	Length	Level of Requirement	Description
callbackReference	common:CallbackReference	N/A	Mandatory	Reference information. Reference information contains the App service address, API name, and correlator ID that a partner provides for the SDP. Based on the reference information, the SDP sends an MO SMS status report notification to the matching App when receiving an MO SMS messages status report from an SMSC. The **callbackReference** parameter is of the CallbackReference type and contains multiple sub-parameters. For details about the CallbackReference type, see Table 3-5.
filterCriteria	xsd:string	50	Optional	Filter keyword. The SDP attempts to match the target user's mobile number from the left to the right based on the filter keyword and sends the corresponding status report to the third party only when the user's mobile number matches the filter keyword.

Parameter	Type	Length	Level of Requirement	Description
				[Example] 1234

NOTE

A root node named **deliveryReceiptSubscription** must be defined for the deliveryReceiptSubscription data structure.

Table 3-5 describes the parameter structure of the CallbackReference type.

Table 3-5 Parameter structure of the CallbackReference type

Parameter	Type	Length	Level of Requirement	Description
notifyURL	xsd:anyURI	512	Mandatory	Service address to which an SMS message status report is sent. [Example] http://10.138.38.139:9080/notify
callbackData	xsd:string	64	Optional	Status report notification correlator ID. When sending an MO SMS message notification to an App, the SDP also sends this parameter value to the App. [Example] 123456
notificationFormat	common:NotificationFormat	N/A	Optional	Encoding format of the body of an MO SMS message sent by the SDP to App. [Enumerated values of **NotificationFormat**] • XML • JSON [Example] XML

3.2.5 Response

The SDP functions as the server, processes Outbound SMS message delivery notification subscriptions messages received from the App, and sends responses to the App.

This topic provides a success response example. If a request fails, the SDP sends an error response that contains an error code. For details about error responses, see API Error Responses.

Example

Response examples in different response body encoding formats are as follows:

- In XML encoding format:

```
HTTP/1.1 201 Created
Content-Type: application/xml; charset=UTF-8
Location:
http://127.0.0.1:8312/1/smsmessaging/outbound/subscriptions/6000130427101719000
0109901
Connection: close
Content-Length: 380
<sms:deliveryReceiptSubscription xmlns:sms="urn:oma:xml:rest:sms:1">
       <callbackReference>
              <notifyURL>http://10.135.178.84:9088/</notifyURL>
              <callbackData>123</callbackData>
              <notificationFormat>xml</notificationFormat>
       </callbackReference>
       <filterCriteria>6513507551001</filterCriteria>

<resourceURL>http://127.0.0.1:8312/1/smsmessaging/outbound/subscriptions/600013
042710171900000109901</filterCriteria>
</sms:deliveryReceiptSubscription>
```

- In JSON encoding format:

```
HTTP/1.1 201 Created
Content-Type: application/json; charset=UTF-8
Location:
http://127.0.0.1:8312/1/smsmessaging/outbound/subscriptions/6000130427101719000
0109901
Connection: close
Content-Length: 363
{
       "deliveryReceiptSubscription": {
              "callbackReference": {
                     "notifyURL": "http://10.135.178.84:9088/",
                     "callbackData": "123",
                     "notificationFormat": "json"
              },
              "filterCriteria": "tel:6513507551001"
              "resourceURL":
"http://127.0.0.1:8312/1/smsmessaging/outbound/subscriptions/600013042710171900
00109901"
       }
}
```

Message header Parameters

Table 3-6 describes parameters in the message header.

Table 3-6 Parameters in the message header

Parameter	Type	Length	Level of Requirement	Description
Content-Type	xsd:string	16	Mandatory	Format in which the response is encoded. The options are as follows: • application/xml • application/json [Example] application/xml
Content-length	int	2	Mandatory	Number of bytes in the response body. [Example] 1024
Location	xsd:string	512	Mandatory	Request URL for the App to unsubscribe from the MO SMS message status report notification function. [Format] http://*serverRoot*/*apiVersion*/smsmessaging/outbound/subscriptions/*subscriptionId* In the format: • serverRoot: root address for the SDP to provide services, which contains the IP address and port number. • apiVersion: API version number. The fixed value is **1**. • *subscriptionId*: unique ID of a subscription request. When the App requests unsubscription, the unsubscription request must contains this parameter. [Example] http://127.0.0.1:8312/1/smsmessaging/outbound/subscriptions/600013042710171900000109901

Message Body Parameters

Table 3-7 describes parameters in the message body.

Table 3-7 Parameters in the message body

Parameter	Type	Length	Level of Requirement	Description
callbackReference	common:CallbackReference	N/A	Mandatory	Reference information. The SDP sends the parameter values in the corresponding request. The **callbackReference** parameter is of the CallbackReference type and contains multiple sub-parameters. For details about the CallbackReference type, see Table 3-8.
filterCriteria	xsd:string	50	Optional	The SDP sends the parameter value if the corresponding request contains this parameter. [Example] 1234
resourceURL	xsd:anyURI	512	Mandatory	Request URL for the App to unsubscribe from the MO SMS message status report notification function. [Format] http://*serverRoot*/*apiVersion*/smsmessaging/outbound/subscriptions/*subscriptionId* In the format: • serverRoot: root address for the SDP to provide services, which contains the IP address and port number. • apiVersion: API version number. The fixed value is **1**. • *subscriptionId*: unique ID of a subscription request. When the App requests unsubscription, the unsubscription request must contains this parameter. [Example] http://127.0.0.1:8312/1/smsmessaging/outbound/subscriptions/600013042710171900001099 01

Table 3-8 describes the parameter structure of the CallbackReference type.

Table 3-8 Parameter structure of the CallbackReference type

Parameter	Type	Length	Level of Requirement	Description
notifyURL	xsd:anyURI	512	Mandatory	Service address to which an SMS message status report is sent. [Example] http://10.138.38.139:9080/notify
callbackData	xsd:string	64	Optional	Status report notification correlator ID. The SDP sends the parameter value if the corresponding request contains this parameter. [Example] 123456
notificationFormat	common:NotificationFormat	N/A	Optional	Encoding format of the body of an MO SMS message sent by the SDP to App. [Enumerated values of **NotificationFormat**] • XML • JSON [Example] XML

3.2.6 Error Codes

Table 3-9 describes Outbound SMS message delivery notification subscriptions error codes that the SDP may return upon an exception. For details about the error codes, see the *SDP Solution Error Code Reference*.

Table 3-9 Outbound SMS message delivery notification subscriptions error codes

Error Code	Description
SVC0002	CallbackReference is null.
	CallbackData of CallbackReference is invalid, the value is %1.
	NotifyUrl is null.
	NotifyUrl %1 is invalid.
	NotifyUrl of CallbackReference is too long, the value is %1.
	NotificationFormat %1 is invalid.

Error Code	Description
	%1 is null.
SVC0901	UserName is null!
	UserName %1 is invalid!
	Service ID is null!
	Service ID %1 is invalid!
SVC0905	Authentication Failed, cause by SP,because of timestamp is not valid.

3.3 Outbound SMS message requests

3.3.1 Function

The App (functioning as the client) invokes the Outbound SMS message requests API to send SMS messages to the SDP (functioning as the server).

Partners must code the App based on the API field requirements so that the App can send correct requests to the SDP. The SDP send a response within 60 seconds by default.

3.3.2 Request Method

Set it to **POST**.

3.3.3 Request URI

The request URI is the destination URI of Outbound SMS message requests requests sent by the App to the SDP. The URI is provided by the SDP in the following format:

http://*IP*:*Port*/*apiVersion*/smsmessaging/outbound/*senderAddress*/requests

In the format, **IP** and **Port** indicate the service IP address and ParlayREST port number of the API provided by the SDP (Contact carriers to obtain the IP address and port number), *senderAddress* indicates the access code planned and provided by the carriers, and *apiVersion* has a fixed value of **1**.

3.3.4 Request

The App functions as the client and sends requests to the SDP.

Example

Request examples in different request body encoding formats are as follows:

- In XML encoding format:

```
POST /1/smsmessaging/outbound/861234/requests HTTP/1.1
Authorization: WSSE realm="SDP",profile="UsernameToken"
X-WSSE: UsernameToken
Username="35000001",PasswordDigest="RnafRPVpUfWp7I2IQnHZxkjh+fc=",Nonce="201304
2718194900001", Created="2013-04-27T18:19:49Z"
X-RequestHeader: request
ServiceId="3500001000001",FA="tel:12312312123",PresentId="12132131",LinkId="121
32131"
Accept-Encoding: gzip,deflate
Accept: application/xml
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.135.178.84:9088
Content-Type: application/xml; charset=UTF-8
Content-Length: 758
<?xml version="1.0" encoding="UTF-8"?>
<sms:outboundSMSMessageRequest xmlns:sms="urn:oma:xml:rest:sms:1">
       <address>6513507551001</address>
       <senderAddress>861111</senderAddress>
       <senderName>1234</senderName>
       <receiptRequest>
              <notifyURL>http://10.135.178.84:9088/</notifyURL>
              <callbackData>12345</callbackData>
              <notificationFormat>xml</notificationFormat>
       </receiptRequest>
       <outboundSMSTextMessage>
              <message>TESTTTTTTTTTT......</message>
       </outboundSMSTextMessage>
       <clientCorrelator>67893</clientCorrelator>
</sms:outboundSMSMessageRequest>
```

- In JSON encoding format:

```
POST /1/smsmessaging/outbound/861111/requests HTTP/1.1
Authorization: WSSE realm="SDP",profile="UsernameToken"
X-WSSE: UsernameToken Username="35000001",
PasswordDigest="UM1e7ksWFVVet89+B/wLn3cv3dI=", Nonce="2013042718231300001",
Created="2013-04-27T18:23:13Z"
X-RequestHeader: request
ServiceId="3500001000001",FA="tel:1350000001",PresentId="12132131",LinkId="1213
2131"
Accept-Encoding: gzip,deflate
Accept: application/json
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.135.178.84:9088
Content-Type: application/json; charset=UTF-8
Content-Length: 823
{
   "outboundSMSMessageRequest":
   {
       "address":
       [
          "6513507551001"
       ],
       "senderAddress":"1234",
       "senderName":"test",
       "receiptRequest":
       {
```

```
            "notifyURL":"http://10.135.178.84:9088/",
            "callbackData":"123",
            "notificationFormat":"json"
        },
        "outboundSMSTextMessage":
        {
            "message":"TESTTTTTTTTTT......"
        }
    }
}
```

- In URL encoding format:

```
POST /1/smsmessaging/outbound/861111/requests HTTP/1.1
Authorization: WSSE realm="SDP",profile="UsernameToken"
X-WSSE: UsernameToken Username="35000001",
PasswordDigest="UM1e7ksWFVVet89+B/wLn3cv3dI=", Nonce="2013042718231300001",
Created="2013-04-27T18:23:13Z"
X-RequestHeader: request
ServiceId="3500001000001",FA="tel:1350000001",PresentId="12132131",LinkId="1213
2131"
Accept-Encoding: gzip,deflate
Accept: application/json
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.135.178.84:9088
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Content-Length: 184
address=tel:13500000991&address=tel:13500000992&senderName=test&senderAddress=1
111&message=Hello%20World&notifyURL=http://10.135.165.4/notify&notificationForm
at=XML&callbackData=123456
```

Message Header Parameters

Table 3-10 describes parameters in the message header.

Table 3-10 Parameters in the message header

Header Field	Parameter	Type	Length	Level of Requirement	Description
Accept	N/A	String	16	Optional	Format in which responses are encoded. The options are as follows: • application/xml • application/json When this parameter is not transmitted, the SDP uses the **application/xml** mode to encode responses by default. [Example] application/xml
Content-Ty	N/A	String	33	Mandatory	Format in which requests are encoded.

Header Field	Parameter	Type	Length	Level of Requirement	Description
pe					• application/xml • application/json • application/x-www-form-urlencoded [Example] application/xml
Content-length	N/A	int	2	Mandatory	Number of bytes in the request body. [Example] 1024
Authorization	/	String	10	Mandatory	Authentication mode. The SDP uses the WSSE authentication mode, in which the **realm** and **profile** parameters are involved. **NOTE** The **Authorization** value indicates the policy used by the SDP to perform authentication at the application layer. The SDP authenticates a third-party in WSSE UsernameToken mode. [Example] WSSE
	realm	String	20	Mandatory	Party that performs authentication. Set it to **SDP**. [Example] SDP
	profile	String	20	Mandatory	Third-party access mechanism. Set it to **UsernameToken**. [Example] UsernameToken
X-WSSE	/	String	30	Mandatory	WSSE authentication flag. To enable the SDP to perform authentication at the application layer in the WSSE UsernameToken mode, set this parameter to **UsernameToken**. [Example] UsernameToken
	Username	String	21	Mandatory	Partner ID. The ID is automatically allocated by the SDP to partners after successful registration. To obtain the ID: • An SP can log in to the SDP management portal and query account information, or log in

Header Field	Parameter	Type	Length	Level of Requirement	Description
					to the mailbox used for registration and view the email notification received after successful registration. • A Developer can log in to the Developer Portal and query account information, or log in to the mailbox used for registration and view the email notification received after successful registration. • An Enterprise must contact the carrier. [Example] 000201
	Password Digest	String	100	Mandatory	Authentication key for the SDP to authenticate partners. The value is a character string encrypted from **Nonce** + **Created** + **Password** by SHA-1 and Base64 in sequence. The encryption formula is as follows: PasswordDigest = Base64(SHA-1(Nonce + Created + Password)). In the formula: • **Nonce** and **Created**: random number and its generation time. • **Password**: password for partners to access the SDP. – An SP can obtain the password from the email notification received after successful registration. – A Developer can log in to the Developer Portal, choose **Member Center** > **Account** > **Registration Information** > **Invoke Password**, and set the password. – An Enterprise must contact the carrier to obtain the password. [Example] e6434ef249df55c7a21a0b45758a39bb
	Nonce	String	30	Mandatory	Random number generated by an App when sending a message. [Example] 66C92B11FF8A425FB8D4CCFE0ED9ED1F
	Created	String	20	Mandatory	UTC time when the **Nonce** value is generated. [Format] yyyy-MM-dd'T'HH:mm:ss'Z' [Example]

Header Field	Parameter	Type	Length	Level of Requirement	Description
					2011-03-22T13:47:56Z
X-RequestHeader	/	String	10	Mandatory	Type of the message sent by a third party. Set it to **request**. [Example] request
	ServiceId	String	21	Conditional	Service ID. The ID is automatically allocated by the SDP to services after successful release. An SP can log in to the SDP management portal and query service information for the ID. This parameter is mandatory in a request sent by an SP and can be left empty in a request sent by a Developer or an Enterprise. [Example] 35000001000001
	LinkId	xsd:string	20	Conditional	Service order ID. The ID is automatically generated by the SDP when a user orders a service in the SDP. This parameter is mandatory during on-demand service delivery by SMS message. The SDP sends the value to SPs as follows in different scenarios: • Invokes the ServiceOnDemand API to send the value when a user orders a service on the SDP portals. • Invokes the Client notification about inbound SMS message API to send the value when a user orders a service by sending an SMS message. [Example] 12345678901111
	PresentId	xsd:string	15	Conditional	Service gift ID. The ID is automatically generated by the SDP when a user sends a service to another user as a gift on the SDP. This parameter is mandatory in an SP's request for sending an SMS message to a gift recipient, and can be left empty in an Enterprise's or a Developer's request. The SDP invokes the assignPresentToUser API to

Header Field	Parameter	Type	Length	Level of Requirement	Description
					send the value to SPs. [Example] 22345678901113
	FA	xsd:string	30	Conditional	Mobile number of the charged party. When sending a single message, set **FA** to the value of mobile number of the charged party. When sending a group message, you can leave the **FA** field empty. Charging is executed based on the **address** field. [Example] 12345678910

Message Body Parameters

Table 3-11 describes the parameter structure of the outboundSMSMessageRequest type in the message body.

Table 3-11 Parameter structure of the outboundSMSMessageRequest type

Parameter	Type	Length	Level of Requirement	Description
address	xsd:anyURI[1..unbounded]	30	Mandatory	Mobile number of the message recipient. [Format] tel:*[Prefix][Country code][Mobile number]* In the format, *[Prefix]* and *[Country code]* are optional. The value of *[Prefix]*, if contained, can be **+**, **+0**, **+00**, **0**, or **00**. **NOTE** If the mobile number is a fake ID, the value consists of 15 characters and is in the format **f** + *a digit string*. [Example] tel:8612312345678
senderAddress	xsd:anyURI	20	Mandatory	Sender address. Set it to the access code allocated by the carrier. • In an SP's request, the value is a service access code obtained

Parameter	Type	Length	Level of Requirement	Description
				from carriers before service release. • In an Enterprise's or a Developer's request, the value is an access code allocated by carriers during capability product purchase. [Example] 321123
senderName	xsd:string	21	Optional	Name of the message sender, which is displayed on users' terminals. [Example] test
receiptRequest	Ocommon:CallbackReference	N/A	Optional	Information required for status report notification. The App sends the App URL, API name, and correlator ID information to the SDP, which then uses the information for status report notification. The **receiptRequest** parameter is of the CallbackReference type and contains multiple sub-parameters. For details about the CallbackReference type, see Table 3-12.
outboundSMSTextMessage	OutboundSMSTextMessage	N/A	Mandatory	Content of the SMS message to be sent. The **outboundSMSTextMessage** parameter is of the OutboundSMSTextMessage type and contains multiple sub-parameters. For details about the OutboundSMSTextMessage type, see Table 3-13.

NOTE
A root node named **outboundSMSMessageRequest** must be defined for the OutboundSMSTextMessage type.

Table 3-12 describes the parameter structure of the CallbackReference type.

Table 3-12 Parameter structure of the CallbackReference type

Parameter	Type	Length	Level of Requirement	Description
notifyURL	xsd:anyURI	512	Mandatory	URL of the App for receiving status reports. [Example] http://10.138.38.139:9080/notify
callbackData	xsd:string	64	Optional	Status report correlator ID. When sending a status report to the App, the SDP also sends this parameter value to the App. The Appfinds the MT SMS message that corresponds to the status report based on this parameter value. [Example] 1234567
notificationFormat	xsd:string	4	Optional	Encoding format of the message body that contains the status report. • XML • JSON [Example] JSON

Table 3-13 describes the parameter structure of the OutboundSMSTextMessage type.

Table 3-13 Parameter structure of the OutboundSMSTextMessage type

Parameter	Type	Length	Level of Requirement	Description
message	xsd:string	700	Mandatory	Content of the SMS message to be sent. When the length of an SMS message exceeds the maximum length (160 GSM 7-bit characters or 70 Unicode characters) supported by the short message service center (SMSC), the message is split into multiple sub messages. [Example] Hello World.

3.3.5 Response

The SDP functions as the server, processes requests received from the App, and sends responses to the App.

This topic provides a success response example and describes parameters in the response. If a request fails, the SDP sends an error response that contains an error code. For details about error responses, see API Error Responses.

Example

Response examples in different response body encoding formats are as follows:

- In XML encoding format:

```
HTTP/1.1 201 Created
Content-Type: application/xml; charset=UTF-8
Location:
http://127.0.0.1:8312/1/smsmessaging/outbound/861234/requests/600001200401130427103027000041
Connection: close
Content-Length: 197
<?xml version="1.0"
encoding="utf-8" ?><resourceReference><resourceURL>http://127.0.0.1:8312/1/smsmessaging/outbound/861234/requests/600001200401130427103027000041</resourceURL></resourceReference>
```

- In JSON encoding format:

```
HTTP/1.1 201 Created
Content-Type: application/json; charset=UTF-8
Location:
http://127.0.0.1:8312/1/smsmessaging/outbound/861111/requests/600001200301130427103354000071
Connection: close
Content-Length: 146
{
      "resourceReference": {
            "resourceURL":
"http://127.0.0.1:8312/1/smsmessaging/outbound/861111/requests/600001200301130427103354000071"
      }
```

Message header Parameters

Table 3-14 describes parameters in the message header.

Table 3-14 Parameters in the message header

Parameter	Type	Length	Level of Requirement	Description
Conte	xsd:strin	16	Mandat	Format in which the response is encoded. The

Parameter	Type	Length	Level of Requirement	Description
nt-Type	g		ory	options are as follows: • application/xml • application/json [Example] application/xml
Content-length	int	2	Mandatory	Number of bytes in the response body. [Example] 1024
Location	xsd:string	512	Mandatory	Message redirected-to URL sent by the SDP. [Format] http://*serverRoot*/*apiVersion*/smsmessaging/outbound/*senderAddress*/requests/*requestId* In the format: • *serverRoot*: service root address provided by the SDP. The format is *IP*:*Port*. • *apiVersion*: The value must be **1**. • *senderAddress*: access code. • *requestId*: unique ID sent by the SDP during SMS message delivery. App can obtain the status report of the delivered message based on the ID. [Example] http://10.137.213.41:8312/1/smsmessaging/outbound/1111/requests/1000012002011109190727230001 31

Message Body Parameters

Table 3-15 describes parameter structure of the resourceReference type in the message body.

Table 3-15 Parameter structure of the resourceReference type

Parameter	Type	Length	Level of Requirement	Description
resourceURL	xsd:anyURI	512	Mandatory	URL that addresses a resource. [Format] http://{*serverRoot*}/*apiVersion*/smsmessaging/outbound/*senderAddress*/requests/*requestId*

Parameter	Type	Length	Level of Requirement	Description
				. In the format: • *serverRoot*: service root address provided by the SDP. The format is *IP*:*Port*. • *apiVersion*: The value must be **1**. • *senderAddress*: access code. • *requestId*: unique ID sent by the SDP during SMS message delivery. App can obtain the status report of the delivered message based on the ID. [Example] http://10.137.213.41:8312/1/smsmessaging/outbound/1111/requests/100001200201110919072723000131

3.3.6 Error Codes

Table 3-16 describes Outbound SMS message requests error codes that the SDP may return upon an exception. For details about the error codes, see the *SDP Solution Error Code Reference*.

Table 3-16 Outbound SMS message requests error codes

Error Code	Description
SVC0001	The message has been licence controled.
SVC0002	Message contains sensitive word %1.
	SenderName or senderAddress is unknown!
	CallbackReference is null.
	CallbackData of CallbackReference is invalid, the value is %1.
	NotifyUrl is null.
SVC0280	Message too long.SLA level is %1, maximum length is %2 characters, actual length is %3 characters.
SVC0901	UserName is null!
	UserName %1 is invalid!
SVC0904	Destination address %1 have duplicate value.
SVC0905	Authentication Failed, cause by SP,because of timestamp is not valid.

Error Code	Description
	Rest FA is null.
	Rest FA %1 is invaild.
POL0900	MultiSubmit not support.
POL0003	Too many addresses. SLA level is %1, maximum number of addresses is %2, actual number of addresses is %3.
POL0904	SP level gross control not pass.

3.4 Client notification SMS message delivery status

3.4.1 Function

The SDP (functioning as the client) invokes the Client notification SMS message delivery status API to send status reports to the App (functioning as the server).

The App invokes the 3.3 Outbound SMS message requests API to send SMS messages to users. Requests of the 3.3 Outbound SMS message requests API contain the status report receiving address. After the SDP sends a request to the SMSC, the SMSC sends a status report. The SDP receives the status report and invokes the Client notification SMS message delivery status API to send the status report to the App.The App must send a response to the SDP within 30 seconds. If the status report fails to be sent, the SDP does not resend it.

Partners must code the App based on the API field requirements so that the App can correctly parse and respond to requests received from the SDP.

3.4.2 Request Method

Set it to **POST**.

3.4.3 Request URI

The request URI is the destination URI of Client notification SMS message delivery status request messages sent by the SDP to the App. The URI is defined by the App.

3.4.4 Request

The SDP functions as the client and sends a request to the App to send status reports.

Example

Request examples in different request body encoding formats are as follows:

- In XML encoding format:

```
POST / HTTP/1.1
Authorization: WSSE realm="SDP",profile="UsernameToken"
X-WSSE: UsernameToken Username="35000001",
```

```
PasswordDigest="u6b6k1L3oBazbVPHfb2fJpQVbOQ=", Nonce="2013042718341400001",
Created="2013-04-27T18:34:14Z"
X-NotifyHeader: notify TransId="5040210000012072319590031959O3"
Content-Type: application/xml; charset=UTF-8
Host: 10.135.178.84:9088
Connection: close
Content-Length: 317
<?xml version="1.0" encoding="utf-8" ?>
<ns1:deliveryInfoNotification xmlns:ns1="urn:oma:xml:rest:sms:1">
  <callbackData>123</callbackData>
  <deliveryInfo>
    <address>8612312312312</address>
    <deliveryStatus>DeliveredToTerminal</deliveryStatus>
  </deliveryInfo>
  </ns1:deliveryInfoNotification>
```

- In JSON encoding format:

```
POST / HTTP/1.1
Authorization: WSSE realm="SDP",profile="UsernameToken"
X-WSSE: UsernameToken Username="35000001",
PasswordDigest="u6b6k1L3oBazbVPHfb2fJpQVbOQ=", Nonce="2013042718341400001",
Created="2013-04-27T18:34:14Z"
X-NotifyHeader: notify TransId="5040210000012072319590031959O3"
Content-Type: application/json; charset=UTF-8
Host: 10.135.178.84:9088
Connection: close
Content-Length: 221
{
       "deliveryInfoNotification": {
              "callbackData": "123",
              "deliveryInfo": [{
                     "address": "8612312312312",
                     "deliveryStatus": "DeliveredToTerminal"
              }]
       }
}
```

Message Header Parameters

Table 3-17 describes parameters in the message header.

Table 3-17 Parameters in the message header

Header Field	Parameter	Type	Length	Level of Requirement	Description
Content-Type	N/A	String	16	Mandatory	Format in which responses are encoded. The options are as follows: • application/xml • application/json [Example]

Header Field	Parameter	Type	Length	Level of Requirement	Description
					application/xml
Content-length	N/A	int	2	Mandatory	Number of bytes in the request body. [Example] 1024
Authorization	/	String	10	Mandatory	Authentication mode. The SDP uses the WSSE authentication mode, in which the **realm** and **profile** parameters are involved. **NOTE** The **Authorization** value indicates the policy used by the SDP to perform authentication at the application layer. The SDP authenticates a third-party in WSSE UsernameToken mode. [Example] WSSE
	realm	String	20	Mandatory	Set it to **SDP**. [Example] SDP
	profile	String	20	Mandatory	SDP access mechanism. Set it to **UsernameToken**. [Example] UsernameToken
X-WSSE	/	String	30	Mandatory	WSSE authentication flag. To enable the SDP to perform authentication at the application layer in the WSSE UsernameToken mode, set this parameter to **UsernameToken**. [Example] UsernameToken
	Username	String	21	Mandatory	Reverse authentication ID for the App to authenticate the SDP. The ID is set by SPs during registration. An SP can log in to the SDP management portal and query account information for the ID. This parameter is mandatory in a request sent to an SP who has configured authentication information during registration. This parameter can be left empty in a request sent to a Developer, an Enterprise, or an SP who does not configure authentication information.

Header Field	Parameter	Type	Length	Level of Requirement	Description
					[Example] sdp
	Password Digest	String	100	Mandatory	Reverse authentication key for the App to authenticate the SDP. The value is a character string encrypted from **Nonce** + **Created** + **Password** by SHA-1 and Base64 in sequence. The encryption formula is as follows: PasswordDigest = Base64(SHA-1(Nonce + Created + Password)). In the formula: • **Nonce** and **Created**: random number and its generation time. • **Password**: access password allocated by an SP to the SDP. An SP can obtain the password from the email notification received after successful registration. This parameter is mandatory in a request sent to an SP who has configured authentication information during registration. This parameter can be left empty in a request sent to a Developer, an Enterprise, or an SP who does not configure authentication information. [Example] 206D88BB7F3D154B130DD6E1E0B8828B
	Nonce	String	30	Mandatory	Random number generated by SDP when sending a message. [Example] 66C92B11FF8A425FB8D4CCFE0ED9ED1F
	Created	String	20	Mandatory	UTC time when the **Nonce** value is generated. [Format] yyyy-MM-dd'T'HH:mm:ss'Z' [Example] 2011-03-22T13:47:56Z
X-NotifyHeader	/	String	10	Mandatory	Type of the message sent by the SDP. Set it to **notify**. [Example] notify
	TransId	String	30	Mandatory	Transaction ID. The ID is automatically generated by the SDP and is used only to trace messages during the SDP

Header Field	Parameter	Type	Length	Level of Requirement	Description
					commissioning. The App ignores this parameter. [Example] 10000120010111062302172100011

Message Body Parameters

Table 3-18 describes the parameter structure of the deliveryInfoNotification type in the message body.

Table 3-18 parameter structure of the deliveryInfoNotification type

Parameter	Type	Length	Level of Requirement	Description
callbackData	xsd:string	64	Optional	Callback data. If a request sent by the App to subscribe to status report notification or send an SMS message contains this parameter, the SDP sends this parameter to the App when sending the corresponding status report. [Example] 1234578
deliveryInfo	DeliveryInfo[1...unbounded]	N/A	Mandatory	Status description. The **deliveryInfo** parameter is of the DeliveryInfo type and contains multiple sub-parameters. For details about the DeliveryInfo type, see Table 3-19.

NOTE
A root node named **deliveryInfoNotification** must be defined for the DeliveryInfoNotification data structure.

Table 3-19 describes the parameter structure of the DeliveryInfo type.

Table 3-19 Parameter structure of the DeliveryInfo type

Parameter	Type	Length	Level of Requirement	Description
address	xsd:anyURI	30	Mandatory	Mobile number of the sender. [Format] tel:*[Prefix][Country code][Mobile number]* In the format, *[Prefix]* and *[Country code]* are optional. The value of *[Prefix]* can be **+**, **+0**, **+00**, **0**, or **00**. [Example] tel:8612312345678
deliveryStatus	DeliveryStatus	40	Mandatory	Status description. [Enumerated values of **DeliveryStatus**] • **DeliveredToTerminal**: The message has been successfully delivered to the terminal. • **DeliveryImpossible**: The message fails to be delivered because of a network error. • **DeliveryNotificationNotSupported**: The SMSC does not provide the function of sending status reports. The SDP constructs status reports. [Example] DeliveredToTerminal

3.4.5 Response

The App functions as the server, processes the request messages received from the SDP, and sends the response messages to the SDP.

The response is constructed based on the SDP specification by the partner that provides the App.

Example

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

3.4.6 Error Codes

The App returns error codes to the SDP when an exception occurs in response to Client notification SMS message delivery status requests. The error codes are defined by partners.

3.5 Individual outbound SMS message delivery notification subscription

3.5.1 Function

The App (functioning as the client) invokes the Individual outbound SMS message delivery notification subscription API to unsubscribe from MO status report notification on the SDP (functioning as the server). This API is invoked generally when the App does not require MO status report notification.

After MO SMS message status report notification is unsubscribed, the SDP does not send MO SMS messages status report received from users to the App.

Partners must code the App based on the API field requirements so that the App can send correct requests to the SDP. The SDP send a response within 60 seconds by default.

3.5.2 Request Method

Set it to **DELETE**.

3.5.3 Request URI

The request URI is the destination URI of Individual outbound SMS message delivery notification subscription requests sent by the App to the SDP. The URI is provided by the SDP in the following format:

http://*IP*:*Port*/*apiVersion*/smsmessaging/outbound/*senderAddress*/subscriptions/*subscriptionId*

In the format, *IP* and *Port* indicate the service IP address and ParlayREST port number of the API provided by the SDP, *senderAddress* indicates the access code planned and provided by carriers (Contact carriers to obtain the IP address, port number and access code), *apiVersion* has a fixed value of **1**, and *subscriptionId* indicates the unique subscription ID, which must be the same as the value of *subscriptionId* in the corresponding Outbound SMS message delivery notification subscriptions response sent by the SDP.

3.5.4 Request

The App functions as the client and sends a Individual outbound SMS message delivery notification subscription request to the SDP to unsubscribe from MO SMS message status report notification.

Example

```
DELETE /1/smsmessaging/outbound/861111/subscriptions/600013042710171900000109901
HTTP/1.1
Authorization: WSSE realm="SDP",profile="UsernameToken"
X-WSSE: UsernameToken
```

```
Username="35000001",PasswordDigest="qZlMyqTTNF4HPv7E8nYrAcwZfyc=",Nonce="201304271
8064200001", Created="2013-04-27T18:06:42Z"
X-RequestHeader: request ServiceId="35000001000001"
Accept-Encoding: gzip,deflate
Accept:
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.137.213.125:14312
Content-Type: application/json; charset=UTF-8
Content-Length: 0
```

Message Header Parameters

Table 3-20 describes parameters in the message header.

Table 3-20 Parameters in the message header

Header Field	Parameter	Type	Length	Level of Requirement	Description
Accept	N/A	String	16	Optional	Format in which responses are encoded. The options are as follows: • application/xml • application/json When this parameter is not transmitted, the SDP uses the **application/xml** mode to encode responses by default. [Example] application/xml
Content-Type	N/A	String	33	Mandatory	Format in which requests are encoded. • application/xml • application/json • application/x-www-form-urlencoded [Example] application/xml
Content-length	N/A	int	2	Mandatory	Number of bytes in the request body. [Example] 1024
Authorization	/	String	10	Mandatory	Authentication mode. The SDP uses the WSSE authentication mode, in which the **realm** and **profile** parameters are involved. **NOTE** The **Authorization** value indicates the policy used by the SDP to perform authentication at the application

Header Field	Parameter	Type	Length	Level of Requirement	Description
					layer. The SDP authenticates a third-party in WSSE UsernameToken mode. [Example] WSSE
	realm	String	20	Mandatory	Party that performs authentication. Set it to **SDP**. [Example] SDP
	profile	String	20	Mandatory	Third-party access mechanism. Set it to **UsernameToken**. [Example] UsernameToken
X-WSSE	/	String	30	Mandatory	WSSE authentication flag. To enable the SDP to perform authentication at the application layer in the WSSE UsernameToken mode, set this parameter to **UsernameToken**. [Example] UsernameToken
	Username	String	21	Mandatory	Partner ID. The ID is automatically allocated by the SDP to partners after successful registration. To obtain the ID: • An SP can log in to the SDP management portal and query account information, or log in to the mailbox used for registration and view the email notification received after successful registration. • A Developer can log in to the Developer Portal and query account information, or log in to the mailbox used for registration and view the email notification received after successful registration. • An Enterprise must contact the carrier. [Example] 000201
	PasswordDigest	String	100	Mandatory	Authentication key for the SDP to authenticate partners. The value is a character string encrypted from **Nonce** + **Created** + **Password** by SHA-1 and Base64 in sequence. The encryption formula is as

Header Field	Parameter	Type	Length	Level of Requirement	Description
					follows: PasswordDigest = Base64(SHA-1(Nonce + Created + Password)). In the formula: • **Nonce** and **Created**: random number and its generation time. • **Password**: password for partners to access the SDP. –An SP can obtain the password from the email notification received after successful registration. –A Developer can log in to the Developer Portal, choose **Member Center** > **Account** > **Registration Information** > **Invoke Password**, and set the password. –An Enterprise must contact the carrier to obtain the password. [Example] e6434ef249df55c7a21a0b45758a39bb
	Nonce	String	30	Mandatory	Random number generated by an App when sending a message. [Example] 66C92B11FF8A425FB8D4CCFE0ED9ED1F
	Created	String	20	Mandatory	UTC time when the **Nonce** value is generated. [Format] yyyy-MM-dd'T'HH:mm:ss'Z' [Example] 2011-03-22T13:47:56Z
X-RequestHeader	/	String	10	Mandatory	Type of the message sent by a third party. Set it to **request**. [Example] request
	ServiceId	String	21	Conditional	Service ID. The ID is automatically allocated by the SDP to services after successful release. An SP can log in to the SDP management portal and query service information for the ID. This parameter is mandatory in a request sent by an SP and can be left empty in a request sent by a Developer or an Enterprise. [Example]

Header Field	Parameter	Type	Length	Level of Requirement	Description
					35000001000001

Message Body Parameters

None

3.5.5 Response

The SDP functions as the server, processes the Individual outbound SMS message delivery notification subscription requests received from the App, and sends the responses to the App.

This topic provides a success response example. If a request fails, the SDP sends an error response that contains an error code. For details about error responses, see API Error Responses.

Example

```
HTTP/1.1 200 OK
Connection: close
Content-Length: 0
```

3.5.6 Error Codes

Table 3-21 describes Individual outbound SMS message delivery notification subscription error codes that the SDP may return upon an exception. For details about the error codes, see the *SDP Solution Error Code Reference*.

Table 3-21 Individual outbound SMS message delivery notification subscription error codes

Error Code	Description
SVC0002	Correlator is null.
	Correlator %1 has a invalid format.
SVC0901	UserName is null!
	UserName %1 is invalid!
	Access authentication or authorization error.
SVC0905	Authentication Failed, cause by SP,because of timestamp is not valid.

3.6 Outbound SMS message delivery status

3.6.1 Function

The App (functioning as the client) invokes the Outbound SMS message delivery status API to obtain status reports from the SDP (functioning as the server).

The App invokes the 3.3 Outbound SMS message requests API to send SMS messages to users. After the SDP sends a request to the SMSC, the SMSC sends a status report. The SDP receives the status report and saves it for 48 hours. When the App invokes the Outbound SMS message delivery status API, the SDP sends status reports within 48 hours to the App.

Partners must code the App based on the API field requirements so that the App can send correct requests to the SDP. The SDP send a response within 60 seconds by default.

3.6.2 Request Method

Set it to **GET**.

3.6.3 Request URI

The request URI is the destination URI of Outbound SMS message delivery status requests sent by the App to the SDP. The URI is provided by the SDP in the following format:

http://*IP*:*Port*/*apiVersion*/smsmessaging/outbound/requests/*requestId*/deliveryInfos

In the format, *IP* and *Port* indicate the service IP address and ParlayREST port number of the API provided by the SDP (Contact carriers to obtain the IP address and port number), **apiVersion** has a fixed value of **1**, and *registrationId* indicates the unique message delivery ID, which must be the same as the value of *requestId* in the corresponding Outbound SMS message requests response sent by the SDP.

3.6.4 Request

The App functions as the client and sends a Outbound SMS message delivery status request to the SDP to obtain status reports.

Example

```
GET /1/smsmessaging/outbound/requests/6000012005011304270955420000001/deliveryInfos
HTTP/1.1
Authorization: WSSE realm="SDP",profile="UsernameToken"
X-WSSE: UsernameToken
Username="35000001",PasswordDigest="j1fxRLQGu2BgS4Y/Zr/XTa8Bn5g=",Nonce="201304271
7450400001", Created="2013-04-27T17:45:04Z"
X-RequestHeader: request
Accept-Encoding: gzip,deflate
Accept: application/xml
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.135.178.84:9088
Content-Type: application/xml; charset=UTF-8
Content-Length: 0
```

Message Header Parameters

Table 3-22 describes parameters in the message header.

Table 3-22 Parameters in the message header

Header Field	Parameter	Type	Length	Level of Requirement	Description
Accept	N/A	String	16	Optional	Format in which responses are encoded. The options are as follows: • application/xml • application/json When this parameter is not transmitted, the SDP uses the **application/xml** mode to encode responses by default. [Example] application/xml
Content-Type	N/A	String	33	Mandatory	Format in which requests are encoded. • application/xml • application/json • application/x-www-form-urlencoded [Example] application/xml
Content-length	N/A	int	2	Mandatory	Number of bytes in the request body. [Example] 1024
Authorization	/	String	10	Mandatory	Authentication mode. The SDP uses the WSSE authentication mode, in which the **realm** and **profile** parameters are involved. **NOTE** The **Authorization** value indicates the policy used by the SDP to perform authentication at the application layer. The SDP authenticates a third-party in WSSE UsernameToken mode. [Example] WSSE
	realm	String	20	Mandatory	Party that performs authentication. Set it to **SDP**. [Example] SDP
	profile	String	20	Mandatory	Third-party access mechanism. Set it to **UsernameToken**.

Header Field	Parameter	Type	Length	Level of Requirement	Description
					[Example] UsernameToken
X-WSSE	/	String	30	Mandatory	WSSE authentication flag. To enable the SDP to perform authentication at the application layer in the WSSE UsernameToken mode, set this parameter to **UsernameToken**. [Example] UsernameToken
	Username	String	21	Mandatory	Partner ID. The ID is automatically allocated by the SDP to partners after successful registration. To obtain the ID: • An SP can log in to the SDP management portal and query account information, or log in to the mailbox used for registration and view the email notification received after successful registration. • A Developer can log in to the Developer Portal and query account information, or log in to the mailbox used for registration and view the email notification received after successful registration. • An Enterprise must contact the carrier. [Example] 000201
	PasswordDigest	String	100	Mandatory	Authentication key for the SDP to authenticate partners. The value is a character string encrypted from **Nonce** + **Created** + **Password** by SHA-1 and Base64 in sequence. The encryption formula is as follows: PasswordDigest = Base64(SHA-1(Nonce + Created + Password)). In the formula: • **Nonce** and **Created**: random number and its generation time. • **Password**: password for partners to access the SDP. –An SP can obtain the password from the email notification received after successful registration. –A Developer can log in to the Developer Portal, choose **Member Center** >

Header Field	Parameter	Type	Length	Level of Requirement	Description
					Account > **Registration Information** > **Invoke Password**, and set the password. – An Enterprise must contact the carrier to obtain the password. [Example] e6434ef249df55c7a21a0b45758a39bb
	Nonce	String	30	Mandatory	Random number generated by an App when sending a message. [Example] 66C92B11FF8A425FB8D4CCFE0ED9ED1F
	Created	String	20	Mandatory	UTC time when the **Nonce** value is generated. [Format] yyyy-MM-dd'T'HH:mm:ss'Z' [Example] 2011-03-22T13:47:56Z
X-RequestHeader	/	String	10	Mandatory	Type of the message sent by a third party. Set it to **request**. [Example] request
	ServiceId	String	21	Conditional	Service ID. The ID is automatically allocated by the SDP to services after successful release. An SP can log in to the SDP management portal and query service information for the ID. This parameter is mandatory in a request sent by an SP and can be left empty in a request sent by a Developer or an Enterprise. [Example] 35000001000001

Message Body Parameters

None

3.6.5 Response

The SDP functions as the server, processes Outbound SMS message delivery status requests received from the App, and sends responses to the App.

This topic provides a success response example and describes parameters in the response. If a request fails, the SDP sends an error response that contains an error code. For details about error responses, see API Error Responses.

Example

Response examples in different response body encoding formats are as follows:

- In XML encoding format:

```
HTTP/1.1 200 OK
Content-Type: application/xml; charset=UTF-8
Connection: close
Content-Length: 314
<?xml version="1.0" encoding="utf-8" ?>
<deliveryInfoList>

<resourceURL>http://127.0.0.1:8312/1/smsmessaging/outbound/requests/60000120050
113042709554200001/deliveryInfos</resourceURL>
  <deliveryInfo>
    <address>6513507551001</address>
    <deliveryStatus>DeliveredToNetwork</deliveryStatus>
  </deliveryInfo>
</deliveryInfoList>
```

- In JSON encoding format:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Connection: close
Content-Length: 256
{
       "deliveryInfoList": {
              "resourceURL":
"http://127.0.0.1:8312/1/smsmessaging/outbound/requests/600001200501130427102712
2000021/deliveryInfos",
              "deliveryInfo": [{
                     "address": "6513507551001",
                     "deliveryStatus": "DeliveryImpossible"
              }]
       }
}
```

Message header Parameters

Table 3-23 describes parameters in the message header.

Table 3-23 Parameters in the message header

Parameter	Type	Length	Level of Requirement	Description
Content-Type	xsd:string	16	Mandatory	Format in which the response is encoded. The options are as follows: • application/xml

Parameter	Type	Length	Level of Requirement	Description
				• application/json [Example] application/xml
Content-length	int	2	Mandatory	Number of bytes in the response body. [Example] 1024

Message Body Parameters

Table 3-24 describes parameters in the message body.

Table 3-24 Parameters in the message body

Parameter	Type	Length	Level of Requirement	Description
resourceURL	xsd:anyURI	512	Mandatory	URL for the App to obtain the status report from the SDP. [Example] http://10.137.213.41:40312/1/smsmessaging/outbound/requests/10000120020111091907272300013 1/deliveryInfos
deliveryInfo	DeliveryInfo[1..unbounded]	N/A	Optional	Status description. The **deliveryInfo** parameter is of the DeliveryInfo type and contains multiple sub-parameters. For details about the DeliveryInfo type, see Table 3-25.

Table 3-25 describes the parameter structure of the DeliveryInfo type.

Table 3-25 Parameter structure of the DeliveryInfo type

Parameter	Type	Length	Level of Requirement	Description
address	xsd:anyURI	30	Mandatory	Mobile number of the sender. [Format] tel:*[Prefix][Country code][Mobile*

Parameter	Type	Length	Level of Requirement	Description
				number] In the format, *[Prefix]* and *[Country code]* are optional. The value of *[Prefix]* can be +, **+0**, **+00**, **0**, or **00**. [Example] tel:8612312345678
deliveryStatus	DeliveryStatus	40	Mandatory	Status description. [Enumerated values of **DeliveryStatus**] • **DeliveredToTerminal**: The message has been successfully delivered to the terminal. • **MessageWaiting**: The message is waiting to be delivered. • **DeliveryUncertain**: The message is being delivered. • **DeliveredToNetwork**: The message has been delivered, but the terminal has not sent any status reports. • **DeliveryImpossible**: The message fails to be delivered because of a network error. • **DeliveryNotificationNotSupported**: The SMSC does not provide the function of sending status reports. The SDP constructs status reports. [Example] DeliveredToTerminal

3.6.6 Error Codes

Table 3-26 describes Outbound SMS message delivery status error codes that the SDP may return upon an exception. For details about the error codes, see the *SDP Solution Error Code Reference*.

Table 3-26 Outbound SMS message delivery status error codes

Error Code	Description
SVC0901	UserName is null!

Error Code	Description
	UserName %1 is invalid!
	Service ID is null!
	Service ID %1 is invalid!
	Authentication Failed, cause by SP,because of timestamp expired.
SVC0905	Authentication Failed, cause by SP,because of timestamp is not valid.

4 API Error Responses

4.1 Service Error Response

A service error is caused by service operation exceptions irrelevant to policies. When a service error occurs, the server sends a service error response to the client. This topic provides a service error response example and describes parameters in the response.

Example

```
<requestError>
   <serviceException>
      <messageId>SVC0280</messageId>
      <text>Message too long.SLA level is %1, maximum length is %2 characters, actual
length is %3 characters.</text>
      <variables>System self</variables>
      <variables>1200</variables>
      <variables>1796</variables>
   </serviceException>
</requestError>
```

Parameter Description

Table 4-1 describes parameters in a service error response.

Table 4-1 Parameters in a service error response

Parameter	Type	Level of Requirement	Description
messageId	xsd:string	Mandatory	Result code. [Format] SVC*ABCD* In the format, **SVC** identifies a service error response, and *ABCD* is a number ranging from 0001 to 9999. [Example] SVC0001

Parameter	Type	Level of Requirement	Description
text	xsd:string	Mandatory	Error description. The value can contain the variable *%#* in definition. [Example] Waiting for response timed out, message type is OutwardGetLocReq.
variables	xsd:string [0..unbounded]	Optional	Value of the variable defined in the value of **text**. [Example] OutwardGetLocReq

4.2 Policy Error Response

A policy error is caused by service level agreement (SLA) violation. When a policy error occurs, the server sends a policy error response to the client. This topic provides a policy error response example and describes parameters in the response.

Example

```
<requestError>
   <policyException>
      <messageId>POL0280</messageId>
      <text>Message too long.SLA level is %1, maximum length is %2 characters, actual
length is %3 characters.</text>
      <variables>System self</variables>
      <variables>1200</variables>
      <variables>1796</variables>
   </serviceException>
</requestError>
```

Parameter Description

Table 4-2 describes parameters in a policy error response.

Table 4-2 Parameters in a policy error response

Parameter	Type	Level of Requirement	Description
messageId	xsd:string	Mandatory	Result code. [Format] POL*ABCD* In the format, **POL** identifies a policy error response, and *ABCD* is a number ranging from 0001 to 9999.

Parameter	Type	Level of Requirement	Description
			[Example] POL0006
text	xsd:string	Mandatory	Error description. The value can contain the variable %# in definition. [Example] GroupAddr is not supported
variables	xsd:string [0..unbounded]	Conditional	Value of the variable defined in the value of **text**. [Example] GroupAddr