Webhook and Callback API

LEKAB Communication Systems AB

Version 7.0.34, 2026-10-19

Webhook and Callback API

Introduction . . . 1
1. Webhook and Callback functionality . . . 2
1.1. GET . . . 2
1.1.1. Delivery receipt - DR . . . 2
1.1.2. Mobile originated - MO . . . 3
1.2. POST . . . 3
1.2.1. Delivery receipt - DR . . . 3
1.2.2. Mobile originated - MO . . . 4
1.3. POST_2 . . . 4
1.3.1. Delivery receipt - DR . . . 4
1.3.2. Mobile originated - MO . . . 6
1.4. POSTFORM . . . 6
1.4.1. Mobile originated - MO . . . 6
1.5. SMTP . . . 6
1.5.1. Delivery receipt - DR . . . 6
1.5.2. Mobile originated - MO . . . 7
1.6. SMTPSPLIT . . . 7
1.6.1. Delivery receipt - DR . . . 8
1.6.2. Mobile originated - MO . . . 9
1.7. Statuses . . . 9
1.8. Error details . . . 10
1.8.1. Rejected or blocked . . . 10
1.8.2. Error details from the mobile network . . . 11
2. HTTP callback retry scheme . . . 12
3. Securing the callbacks . . . 13
3.1. Including a token for security . . . 13
3.2. Using the receiving server firewall . . . 13

Introduction

© 2006 - 2024 Lekab Communication Systems AB. Version 7.0.34, 2026-10-19.

Chapter 1. Webhook and Callback functionality

The SMS platform includes functionality for callback (pushing, event-driven reporting) of inbound messages (MO) and delivery receipts (DR) to customer environments.

The following callbacks are available:

- **DR** - Delivery receipt
- **MO** - Mobile originated SMS

This is in addition to, and complementing the existing polling functionality available through the web services (and through logging in to the web interfaces and looking). The customer will supply a URL where their service will be listening for callback from this functionality. In some of our APIs, an individual URL for each DR can be supplied (see the respective API manuals), but in most cases, the DR URL is set on the sending user account. MO URLs are always set on the sending user account.

All the timestamps are from the CET timezone. The customer can specify their preferred timezone in their account settings.

1.1. GET

HTTP GET as UTF-8 URL-encoded URL parameters. Note that space `%20` is replaced with `+`. See https://tools.ietf.org/html/rfc3986#section-2.1 and https://www.w3.org/TR/html401/interact/forms.html#h-17.13.4.1

A callback has succeeded when the customer's server responds with 200 OK.

1.1.1. Delivery receipt - DR

Parameters:

- `id` - Unique message ID generated by the service
- `from` - Sender ID e.g. 46737494333000 or BIGCO or 70303
- `to` - Recipient e.g. 46701234567
- `status` - Status e.g. Delivered (see Statuses section below)
- `statusmessage` - Status details e.g. DELIVRD (see Statuses section below)
- `timestamp` - Formatted timestamp `YYYY-MM-ddTHH:mm:ss` e.g. `2009-02-14T00:31:30`

Example:

```
https://www.example.com/sms/dr/callback?id=1363257&from=70303&to=12345678901&status
=Delivered&statusmessage=DELIVRD&timestamp=2009-02-14T00%3A31%3A30
```

1.1.2. Mobile originated - MO

Parameters:

- `id` - Unique message ID generated by the service
- `from` - Sender ID e.g. 46701234567
- `to` - Recipient e.g. 46737494333000 or a short code like 70303
- `message` - The message
- `timestamp` - Formatted timestamp `YYYY-MM-ddTHH:mm:ss` e.g. `2009-02-14T00:31:30`

Example:

```
https://www.example.com/sms/mo/callback?id=19334&from=12345678900&to=70303&message=
I+am+a+little+teapot&timestamp=2009-02-14T00%3A31%3A30
```

Note that this can also be used for customers that poll the inbox after we send a GET polling-trigger, if they ignore the query parameters.

1.2. POST

HTTP POST as a JSON document in the message body.

A callback has succeeded when the customer's server responds with 200 OK.

1.2.1. Delivery receipt - DR

Parameters in the JSON document:

- `id` - Unique message ID generated by the service
- `sender` - Sender ID e.g. 46737494333000 or BIGCO or 70303
- `recipient` - Recipient e.g. 46701234567
- `status` - Status e.g. Delivered (see Statuses section below)
- `statusmessage` - Status details e.g. DELIVRD (see Statuses section below)
- `conversation` - Conversation ID
- `time` - Formatted timestamp `YYYY-MM-ddTHH:mm:ss` e.g. `2009-02-14T00:31:30`

Example:

```
{
  "id":"1363257",
  "status":"Delivered",
  "statusmessage":"DELIVRD",
  "sender":"70303",
  "recipient":"12345678901",
```

```
  "conversation":"ABC123",
  "time":"2009-02-14T00:31:30"
}
```

1.2.2. Mobile originated - MO

Parameters:

- `id` - Unique message ID generated by the service
- `sender` - Sender ID e.g. 46701234567
- `recipient` - Recipient e.g. 46737494333000 or a short code like 70303
- `message` - The message
- `conversation` - Conversation ID
- `time` - Formatted timestamp `YYYY-MM-ddTHH:mm:ss` e.g. `2009-02-14T00:31:30`

Example:

```
{
  "id":"19334",
  "sender":"12345678900",
  "recipient":"70303",
  "message":"I am a little teapot",
  "conversation":"",
  "time":"2009-02-14T00:31:30"
}
```

1.3. POST_2

HTTP POST as a JSON document in the message body.

A callback has succeeded when the customer's server responds with 200 OK.

This is a much extended version of the `POST` (JSON) callback described above. The old `POST` callback is retained unchanged for legacy integrations. With the `POST_2` callback, we hereby warn that new fields may be added later, and that fields where no content is available will not be present in the JSON. Therefore, deserialization code needs to not stop on new fields or on missing fields.

So far, the incoming/MO callback is identical to the `POST` callback type, so only the receipt/DR callback format is described here.

1.3.1. Delivery receipt - DR

Parameters in the JSON document:

- `id` - Unique message ID generated by the service

- `status` - Status (summary) e.g. Delivered (see Statuses section below)
- `statuscode`- Code for the summary status (2-14, see Statuses section below)
- `statusmessage` - Status details e.g. DELIVRD (see Statuses section below)
- `sender` - Sender ID e.g. 46737494333000 or BIGCO or 70303
- `recipient` - Recipient e.g. 46701234567
- `conversation` - Conversation ID given by the sender echoed here
- `time` - Timestamp in UTC time zone `YYYY-MM-ddTHH:mm:ssZ` e.g. `2009-02-14T00:31:30Z`
- `messagetext`- Text of the message sent (if still available)
- `countrycallingcode` - Calling code of the destination number (billing info)
- `numberofparts` - Number of SMS parts in the message (billing info)
- `costcenter` - Cost center given by the user for splitting bill (billing info)
- `billingid` - Billing id assigned to user or otherwise set by system (billing info)
- `senttime` - Time of sending in UTC time zone e.g. `2009-02-14T00:31:30Z`
- `submittime` - Time of submit to operator in UTC time zone e.g. `2009-02-14T00:31:30Z`
- `updatetime` - Time of last update before done in UTC time zone e.g. `2009-02-14T00:31:30Z`
- `donetime` - Time sending finished in UTC time zone e.g. `2009-02-14T00:31:30Z`
- `errordetails` - Code from the underlying system for support use. See separate chapter below.

Example:


```
{
  "id": "17793542",
  "status": "Delivered",
  "statuscode": 2,
  "statusmessage": "DELIVRD",
  "sender": "BIGCO",
  "recipient": "46706352602",
  "conversation": "",
  "time": "2026-04-08T09:00:43Z",
  "messagetext": "This text comes in the DR callback json",
  "countrycallingcode": "46",
  "numberofparts": "1",
  "costcenter": "DEPT01",
  "billingid": "STOCKHOLM",
  "senttime": "2026-04-08T09:00:33Z",
  "submittime": "2026-04-08T09:00:33Z",
  "updatetime": "2026-04-08T09:00:33Z",
  "donetime": "2026-04-08T09:00:43Z",
  "errordetails": "000"
}
```


Some of these fields are not usually present simultaneously (e.g. costcenter and billingid) and

null fields will not be included in the JSON. Therefore, the deserializing code should not stop on missing fields or on new fields added in later versions.

1.3.2. Mobile originated - MO

So far unchanged from `POST` callback type - subject to change later.

1.4. POSTFORM

HTTP POST UTF-8 URL-encoded in the message body. Note that space `%20` is replaced with `+`. See https://tools.ietf.org/html/rfc3986#section-2.1 and https://www.w3.org/TR/html401/interact/forms.html#h-17.13.4.1

A callback has succeeded when the customer's server responds with 200 OK.

1.4.1. Mobile originated - MO

Parameters:

- `id` - Unique message ID generated by the service
- `from` - Sender ID e.g. 46701234567
- `to` - Recipient e.g. 46737494333000 or a short code like 70303
- `message` - The message
- `timestamp` - Formatted timestamp `YYYY-MM-ddTHH:mm:ss` e.g. `2009-02-14T00:31:30`

Example:

```
id=19334&from=12345678900&to=70303&message=I+am+a+little+teapot&timestamp=2009-02-14T00%3A31%3A30
```

1.5. SMTP

The callback will be sent as a plain text message to an email address of the customer's choice, which is given in the URL field of the settings.

There is no check whether the callback email was successfully delivered.

Note that the callback will be sent as an email in all cases, independent on whether the original SMS was generated from an email, by using an API (REST or SOAP) or from the web portal Send SMS page.

1.5.1. Delivery receipt - DR

Example from sender:

```
no-reply@sms.lekab.com
```

Subject line:

```
VSO delivery notification
```

Message text:

```
VSO delivery notification
------------------------------------------------------------

Message 1041045547 sent to 46701234567
Time: 2021-11-17T10:12:07.911
Subject:

Status : Delivered

------------------------------------------------------------
```

If the message was not generated by email-to-SMS, the Subject field will be empty, while if it was, the Subject line of the original email-to-SMS email will be shown as the Subject.

1.5.2. Mobile originated - MO

Example from sender:

```
46701234567@sms.lekab.com
```

Subject line:

```
SMS from 46701234567 to 46737494333000
```

Message text:

```
Yes

Message received 2021-11-17 10:12:35
```

1.6. SMTPSPLIT

The callback will be sent as a plain text message to an email address which is supposed to have been stored when the user sent an SMS by using the email-to-SMS functionality. This is in

contrast to the SMTP setting (above) which callbacks only to the given email address in the URL field.

To this end, the email-to-SMS functionality administrator must set the email-to-SMS account to "shared user, split callback" so that the sender email will be stored for the purpose of callback.

If the email-to-SMS functionality was not used to send the original SMS, the email address in the URL will be used for callback of receipt and incoming reply. For "cold" incoming messages to a reserved incoming number (short code or long number) the email address in the URL will likewise be used. Note that the callback will be sent as an email in all cases, independent on whether the original SMS was generated from an email, by using an API (REST or SOAP) or from the web portal Send SMS page.

If a blank URL is given, there will be no callback at all, for any messages. Use any dummy URL like "DUMMY" or "X" or "IGNORE" if cold incoming messages and non-email originated SMS receipts and responses are to be ignored, but email originated receipts and two-way responses are to be callbacked to the sender email address.

There is no check whether the callback email was successfully delivered.

1.6.1. Delivery receipt - DR

Example callback email from sender:

```
no-reply@sms.lekab.com
```

Callback email subject line:

```
VSO delivery notification
```

Message text:

```
VSO delivery notification
------------------------------------------------------------

Message 1041295921 sent to 46701234567
Time: 2021-11-17T14:41:54.795
Subject: Important message

Status : Delivered

------------------------------------------------------------
```

If the message was not generated by email-to-SMS, the Subject field will be empty, while if it was, the Subject line of the original email-to-SMS email will be shown as the Subject.

1.6.2. Mobile originated - MO

Example callback email from sender:

```
46701234567@sms.lekab.com
```

Callback email subject line:

```
SMS from 46701234567 to 46737494333000
```

Message text:

```
Yes

Message received 2021-11-17 10:12:35
```

1.7. Statuses

Delivery Receipt (DR) callback messages indicate success or failure of sending an SMS message. Usually, either the message got through to the (turned-on) mobile handset of the recipient: `Delivered`, or there is an error status.

We send DR callback messages when triggered by getting a delivery receipt from an SMSC further on in the distribution chain, with a status message in it. Thus, we do not send DR callbacks when a message moves from one internal queue to another, nor do we send DR callbacks when a message leaves the last internal queue and is sent to the first in the chain of SMSCs delivering the message to the customer's home SMSC. However, as soon as we get a notification of success or failure of the sending, we send a DR callback.

If we do not get notification of either success or failure, we do not know whether the message will eventually reach the recipients, but if takes too long, the message will expire and we (from most operators in most countries) get a message telling us that. If no such message arrives, we will not send any callback, but this is a very rare occurrence.

Most messages are delivered, and most operators then send only "DELIVRD", with no extra comments. Otherwise, the status messages we receive when sending fails differ greatly between operators and countries, and we keep a translation table to map known status messages to one of a small number of statuses. In the DR callback, we show this summary status (e.g. Delivered) and also the status message we got from the distributor chain (e.g. DELIVRD or something longer).

The following list gives the statuses that can occur in a DR callback:

status code	status	Description
2	Delivered	Delivered to the mobile station
3	Deleted	The message was deleted
4	Expired	The message has expired
5	Rejected	The message was rejected by the operator
6	Undeliverable	The message could not be delivered
8	AbsentSubscriber	The subscribers mobile station is switched off
9	UnknownSubscriber	The subscriber is not known
10	InvalidDestination	The destination address is invalid
11	SubscriberError	The mobile station can not receive the message
12	Unknown	The status of the message is unknown
13	Error	Internal error when sending the message
14	Read	Recipient read message (only for non-SMS channel uplift)

1.8. Error details

When a message could not be delivered, the POST_2 delivery receipt may include additional information in the `errordetails` property.

1.8.1. Rejected or blocked

If a message was rejected or blocked before it was submitted to the mobile network, the following information is returned in `errordetails`:

Error details	Description
QUOTA_C	The message was rejected because the quota limit set on the company level has exceeded.
QUOTA_U	The message was rejected because the quota limit set on the sending user has exceeded.
QUOTA_P	The message was rejected because the quota limit for the recipient number prefix has exceeded. The number prefix is usually a country code.
BLOCKED_U	The message was rejected due to a blocking rule configured for the user account.
BLOCKED_C	The message was rejected due to a blocking rule configured on the company level.

Error details	Description
OPT_OUT	The message was rejected because the mobile recipient has previously opted out, for example by sending a STOP keyword.

Note: When messages are rejected due to quota limits, it will be possible to send again when the new period starts (i.e. hour, day, week, month, year). The calling application is responsible for resending.

1.8.2. Error details from the mobile network

If a message failed after it was submitted to the mobile network, the `errordetails` may contain additional information from external systems such as the SMSC or aggregators involved in routing the message to the recipient.

If the SMS message was submitted over the SMPP connection, the `errordetails` may contain additional information according to the SMPP standard. This manual does not include a complete list of all possible content in the error details property because it is forwarded as it was received from the external systems. For a full list of possible SMPP errors, please refer to the Short Message Peer-to-Peer Protocol Specification 3.6 or 5.0.

For example, when `statuscode = 8` meaning `AbsentSubscriber`, the following additional information may appear in `errordetails` explaining why the subscriber was considered absent:

- Absent Subscriber, cause IMSI detached
- Absent Subscriber, cause Roaming restriction
- Absent Subscriber, Deregistered in the HLR for non-GPRS
- Absent Subscriber, MS purged for non-GPRS
- Absent Subscriber, No paging response via the MSC

Note: When messages are delivered successfully, the value '000' is often returned.

Chapter 2. HTTP callback retry scheme

If the server at the customer supplied url does not respond to a callback attempt with `200 OK` within a reasonable time, for the HTTP type callbacks documented above (`GET`, `POST`, `POST_2` and `POSTFORM`), the callback will be retried. The times (not currently settable except in the database) are as follows (distinct for DR and MO). There is a time-to-live after which no more retries are attempted. That time is 2 hours for DR and MO. There is also a base delay for retries and a delay delta for ever increasing delay between retries (up to the time-to-live), with a difference between DR and MO.

```
callbackDrTtl   = 2 hours
callbackDrDelay = 2 minutes
callbackDrDelta = 1 minute
callbackMoTtl   = 2 hours
callbackMoDelay = 2 minutes
callbackMoDelta = 2 minutes
```

This means that DR and MO callback will be attempted up to two hours, with attempts at minutes:

```
DR: 0, 2, 5, 9, 14, 20, 27, ...
MO: 0, 2, 6, 12, 20, 30, 42, ...
```

Chapter 3. Securing the callbacks

HTTPS (encrypted http) is, obviously, the standard method to ensure message integrity in transit. If message origin authentication is needed for the callback functionality, there are two main methods: including a **token** with the callback and using the **firewall** on the receiving server. These can be combined.

3.1. Including a token for security

The callback url set for the user can have a apikey-like url parameter appended, thus:

```
https://server.mycompany.com/webhooks/smscallback?key=ABCD1234
```

It is also possible to send any HTTP headers necessary for authentication by setting them up for the user. This is currently not visible in the web portal, but your system administrator can add them to the database. Each header is characterized by user account id, MO or DR, name and value. Each relevant header will be appended as an HTTP header to the callback HTTP call. Content is up to the user. For example if the user 12345 has a DR header with name `Authorization` and value `ABCD1234`, all callback HTTP calls of delivery receipts to user 12345 will have the HTTP header

```
Authorization: ABCD1234
```

in the HTTP request.

3.2. Using the receiving server firewall

This manual applies to systems in different countries, but a given customer is always registered as a user at one system only. Usually, each system has several IP addresses (redundant servers) from which the callback HTTP/HTTPS calls originate. Our system administrator can provide these IP addresses for your country. That information can be used to lock down the callback url so that it only receives calls from our IP addresses. That is, obviously, done locally in the customer's system, and cannot be done by our system administrators.