

Cellact JSON API - SMS Sending Guide Version 1.3.5

Developer Documentation

Webhook delivery status and REASON code clarification

1. Overview

The Cellact JSON API enables high-performance SMS delivery using a secure HTTP interface. All requests to the sending API use HTTPS, JSON, and Bearer authentication.

Base URL:

<https://la.cellactpro.com/v1/messages>

2. Authentication

Use your API key as a Bearer token.

Authorization: Bearer <API_KEY_ID>

Content-Type: application/json; charset=utf-8

Accept: application/json

3. Send SMS - Endpoint

POST /v1/messages

This endpoint accepts a JSON payload containing message details.

4. Request Body Example

```
{
  "to_s": ["+972520123456", "+972541234567"],
  "text": "Hello from Cellact",
  "sender": "Cellact",
  "webhook_url": "https://my.company.com/hook",
  "ttl": 15,
  "msg_id": "my-custom-id-001"
}
```

- **to_s:** [Mandatory] The recipients list for the message (up to 200).
- **text:** [Mandatory] The message content (up to 1000 characters).
- **sender:** [Mandatory] The message sender (SID).
- **webhook_url:** [Optional] HTTPS URL to which delivery status callbacks will be sent.
- **ttl:** [Optional] Maximum time to keep trying to deliver the message in case of unavailable recipients or other errors.
- **msg_id:** [Optional] Customer-provided message identifier. When supplied, it is returned in delivery status callbacks.

5. Python Example

```
import requests

url = "https://la.cellactpro.com/v1/messages"
values = {
    "sender": "Cellact",
    "to_s": ["+972511111111", "+972522222222"],
    "text": "Hello from Cellact",
    "webhook_url": "https://MyURL.example/hook"
}

headers = {
    "Content-Type": "application/json",
    "Authorization": "Bearer <API_KEY_ID>"
}

response = requests.post(url, json=values, headers=headers)

if response.status_code == 200:
    print("Message sent successfully!")
    print(f"Response: {response.text}")
else:
    print(f"Error sending message. Status code: {response.status_code}")
    print(response.text)
```

6. Success Response Example

```
{
  "result": true,
  "id": "ef60ca0c-540f-4883-abbd-205b6ca9de34"
}
```

7. Failure Response Example

```
{
  "result": false,
  "description": "invalid request",
  "errorCode": 10
}
```

8. Delivery Status Webhook

When `webhook_url` is supplied in the send request, Cellact sends delivery status callbacks to that HTTPS endpoint.

8.1 Callback lifecycle

A single SMS message can generate up to two delivery status callbacks. No more than two delivery status callbacks are sent for the same message.

For a successful delivery flow, the callbacks are:

- **mt_ok:** First callback. Indicates the initial successful message processing stage. It is not the final delivery confirmation.
- **mt_del:** Second and final callback in a successful flow. Indicates successful delivery.

Successful flow: `mt_ok` -> `mt_del`

Negative flow: `mt_nok` (reason identified by `errorId`) -> `final`

Operator delivery failure flow: `mt_ok` -> `mt_rej` (`errorId` 7001) -> `final`

Callbacks for the same message use the same id, allowing the receiving system to correlate them to the same SMS message. Some negative scenarios can terminate after a single callback; the maximum remains two callbacks per SMS message. The second callback is not necessarily immediate and can arrive several hours later, depending on the configured delivery lifetime and the mobile operator retry process.

8.2 Successful callback examples

First callback - `mt_ok`:

```
{
  "id": "9b291887-8690-48d6-9dcf-f2c9214b3dfa",
  "sender": "Cellact",
  "to": "+972111111111",
  "when": "20260907161442",
  "event": "mt_ok",
  "errorId": "5000",
  "description": "5000",
  "stamp": "1788786880545-1662650000493178"
}
```

Second and final callback - `mt_del`:

```
{
  "id": "9b291887-8690-48d6-9dcf-f2c9214b3dfa",
  "sender": "Cellact",
  "to": "+972111111111",
  "when": "20260907161442",
  "event": "mt_del",
  "errorId": "1000",
  "description": "1000",
  "stamp": "1788786880545-1662650000493178"
}
```

8.3 Negative callback - mt_nok

The mt_nok event is a terminal negative callback. The specific reason is identified by the errorId value. However, negative REASON codes are not necessarily exclusive to mt_nok: depending on where the failure is determined, a negative reason can also be returned with mt_rej. Clients should evaluate event and errorId together rather than hard-code each negative errorId to a single event type.

The Guardian-blocked condition is internally referred to as MT_GNOK and is represented by errorId "2050". For mt_nok terminal cases, no mt_del callback will follow for that message.

Confirmed mt_nok examples currently documented:

Error ID	Meaning
2050	Recipient blocked by Cellact's Guardian blocking mechanism (MT_GNOK condition).
4002	The available balance in the account has been exhausted.
2055	No suitable balance is available for the destination number type. This can occur when sending to a kosher number or an international number when the account does not have the required balance and authorization for that destination type.
2035	The Sender ID is not authorized on the account. The sender must be approved by Cellact Support before it can be used.

Example - recipient blocked by Guardian:

```
{
  "id": "df6e7f7f-8394-4739-a8ed-e01206f29331",
  "sender": "Cellact",
  "to": "+972511111111",
  "when": "20260907165220",
  "event": "mt_nok",
  "errorId": "2050",
  "description": "2050",
  "stamp": "Zc55b950-37b3-48f6-acdf-0e1f88ac0a74"
}
```

8.4 Operator delivery rejection - mt_rej

When the mobile operator cannot deliver the message to the recipient and the configured delivery time and retry attempts are exhausted, Cellact sends a final negative callback with event "mt_rej".

For the confirmed case documented here, errorId "7001" indicates that delivery was not completed after the operator delivery attempts were exhausted. This is a terminal negative status: no mt_del callback will follow for that message.

The mt_rej callback can arrive several hours after the initial mt_ok callback because the message may remain in the operator delivery/retry process for several hours before the final result is returned.

mt_rej can also carry a specific REASON code returned during the operator delivery process, rather than only errorId "7001". For example, when the mobile operator reports that the subscriber is unavailable, the callback can be mt_rej with errorId "1002". Therefore, the receiving system should interpret the event and errorId as a pair and should not assume that a negative errorId can only appear with mt_nok.

Example - operator delivery attempts exhausted:

```
{
  "id": "de90bc27-2787-42d7-abcc-f20145cd1f10",
```

```

"sender": "Cellact",
"to": "+972511111111",
"when": "20260908164927",
"event": "mt_rej",
"errorId": "7001",
"description": "7001",
"stamp": "1788875316097-1203961972516718"
}

```

8.5 Callback fields

Field	Type	Presence	Description
id	string	Required	Cellact message identifier. The same id is used for the callbacks of the same SMS message.
sender	string	Required	Sender ID of the SMS message.
to	string	Required	Recipient phone number.
msg_id	string	Optional	Customer-provided identifier. Returned when msg_id was supplied in the original send request.
when	string	Required	Event timestamp in YYYYMMDDHHmmss format, for example 20260907161442.
event	string	Required	Delivery event. Confirmed values documented here: mt_ok, mt_del, mt_nok, and mt_rej.
errorId	string	Required	Status / REASON code associated with the callback. See Appendix A for the supplied REASON reference. Negative reason codes can be returned with mt_nok or mt_rej depending on where the failure is determined. Confirmed examples include 5000 for mt_ok, 1000 for mt_del, 2050 for a Guardian block, and 7001 for a final operator non-delivery result.

Field	Type	Presence	Description
description	string	Required	Description/status value. In the confirmed examples it contains the same code value as errorId.
stamp	string	Required	System-generated tracking value. Treat this value as an opaque string; no fixed format should be assumed.

Value types: The documented delivery callback fields above are strings. Optional fields may be omitted when they do not apply. The examples do not define null-valued fields.

8.6 Receiver implementation guidance

- **Correlation:** Use id to associate up to two callbacks with the same SMS message. If msg_id was supplied, it may also be used as a customer-side correlation identifier. Do not expire the correlation record too early: a final mt_rej callback can arrive several hours after mt_ok.
- **State handling:** Treat mt_ok as an intermediate state, mt_del as the final successful delivery state, and mt_nok / mt_rej as terminal negative states. Interpret event and errorId together. A negative REASON code is not necessarily tied to only one event type: depending on whether the failure is determined by Cellact or returned by the mobile operator, the code can be reported with mt_nok or mt_rej. For example, operator-reported subscriber unavailability can be returned as mt_rej with errorId "1002". Confirmed mt_nok examples include 2050 (Guardian block), 4002 (account balance exhausted), 2055 (no suitable balance/authorization for the destination number type), and 2035 (Sender ID not authorized).
- **Schema tolerance:** Do not reject a callback solely because it contains additional fields. Ignore unknown fields unless your integration explicitly uses them.
- **Idempotency:** Implement callback processing so that receiving the same event more than once does not cause duplicate business actions.
- **Parsing:** Do not parse errorId, description, when, or stamp as numeric values. They are documented as strings.

8.7 Payload format and Content-Type

Delivery status callbacks are sent with the HTTP header Content-Type: application/json.

The callback request body is a JSON object. It is not sent as application/x-www-form-urlencoded. The receiving endpoint should accept and parse the request body as JSON.

8.8 Event values covered by this revision

This revision documents the confirmed event values mt_ok, mt_del, mt_nok, and mt_rej. Appendix A adds the supplied REASON reference values. Negative REASON codes should not be validated against a fixed one-to-one event mapping: depending on the processing stage and source of the result, a negative reason may be returned with mt_nok or mt_rej. For example, operator-reported subscriber unavailability

may be returned as mt_rej with errorId "1002". The confirmed operator delivery-exhausted case uses mt_rej with errorId "7001" and may arrive several hours after mt_ok.

9. MO Event Example

The MO (message from the recipient) is set per a virtual number which you can buy for the account.

We will set the virtual number to send the MO (like in the example below) to a predefined URL that points to a page in the customer environment.

```
{
  "company": "Cellact",
  "id": "47b02779-ad5b-4ee4-8a51-5d931b357afe",
  "sender_op": "97250",
  "sender": "+972502888641",
  "to": "+972557000918",
  "when": "20251207090710",
  "content": "Thanks",
  "event": "mo",
  "cp_mo": "cellact_communications"
}
```

10. Revision Notes

Version 1.3.5 expands the delivery status webhook documentation with the supplied REASON reference table and clarifies that negative REASON codes may be returned with mt_rej as well as mt_nok, depending on where the failure is determined.

- **Webhook lifecycle:** Clarified that a message can generate up to two delivery status callbacks.
- **Successful event sequence:** Documented mt_ok followed by mt_del.
- **Callback schema:** Added field types, required/optional status, when format, and the stamp field.
- **Examples:** Replaced the previous single mt_del example with a two-event successful-flow example based on the current callback structure.
- **Receiver guidance:** Added correlation, state handling, schema tolerance, and idempotency recommendations.
- **Content-Type: Confirmed that delivery status callbacks are sent as JSON with Content-Type: application/json and are not sent as application/x-www-form-urlencoded.**
- **mt_nok handling:** Clarified that mt_nok is a terminal negative event whose cause is identified by errorId. Retained the Guardian / MT_GNOK case as errorId "2050".
- **Additional mt_nok error codes:** Added errorId "4002" for exhausted account balance, "2055" for missing suitable balance/authorization for the destination number type (including kosher or international destinations), and "2035" for a Sender ID that is not authorized on the account.
- **Operator delivery rejection:** Added the confirmed mt_rej callback with errorId "7001" when the operator cannot deliver the SMS and the configured delivery/retry process is exhausted. Clarified that this final callback may arrive several hours after mt_ok.
- **REASON reference:** Added Appendix A with the supplied REASON values and descriptions, including SMS and other message-service reason codes.
- **mt_rej REASON handling:** Clarified that mt_rej may contain a specific operator-returned REASON code. Example: errorId "1002" when the operator reports that the subscriber is unavailable.

- 4002 documentation note: The previously confirmed webhook description supplied for errorId "4002" is "account balance exhausted". The supplied Appendix source describes 4002 differently ("system did not recognize message's Tariff Prepaid"). This discrepancy is explicitly flagged in Appendix A and should be resolved before treating either wording as a single canonical definition.

Appendix A - REASON Values

The following values are reproduced from the supplied <REASON> reference. In webhook delivery callbacks, the value is carried in errorId. For negative delivery outcomes, do not assume a one-to-one mapping between REASON and event: depending on where the failure is determined, a negative REASON can be returned with mt_nok or mt_rej. For example, when the mobile operator reports that the subscriber is unavailable, Cellact can return event "mt_rej" with errorId "1002".

Scope note: This is an SMS sending guide, but the supplied REASON table also contains values explicitly related to WAP, MMS, MO and email. Those entries are included for reference and are not presented as SMS-only delivery statuses.

REASON	Description
1000	Message delivered
1002	Subscriber is unavailable
1003	Message inbox full
1005	Incorrect subscriber number
1101	Registration request parameters are missing/incorrect
1103	Password not authenticated
1104	Subscriber is already registered with this service
1105	Subscriber is not registered with this service
1106	Pending request in process
1107	Internal connection error
1108	System SQL Connection Error
1140	Invalid SIGI / Subscriber is not registered with this service
1141	TRID or SIGI is missing
2006	Destination is blocked to RB messages
2010	Subscriber is blocked
2013	Sender is invalid
2014	Sender from a different operator cannot be charged
2016	Destination does not support the message's type
2018	Unknown link
2020	Spam filter
2050	Subscriber block in Guardian
2060	Number is blocked for SMS reception

REASON	Description
3001	Account balance has exceeded
3002	Message contains invalid attributes
3004	Validator Error MNP
3007	Night Block feature - account cannot send messages during the predefined hours
4002	Supplied Appendix wording: system did not recognize message's Tariff Prepaid
5000	Message was delivered to the operator
5001	Operator error
5007	Recipient is temporarily unavailable / Device inbox is full
5014	SQL: System connection error
5016	Message size exceeded
5028	Error sending overseas message
6001	Communication error
7000	Subscriber replied to MT message
7001	Message not delivered
8000	Wap text message was opened
8018	MMS message rejected
9000	MO message delivered
9005	Subscriber has insufficient balance
9801	Email message was not delivered
9821	Recipient's Server is not available
9831	Message declined by recipient
9840	Recipient is unavailable / Mail inbox is full
9845	Recipient is Not Found
9850	Invalid Email Address

Documentation discrepancy - REASON 4002: Earlier confirmed webhook behavior supplied for this integration defines errorId "4002" as "account balance exhausted". The supplied Appendix <REASON> table describes 4002 as "system did not recognize message's Tariff Prepaid". These two descriptions conflict. The webhook-specific confirmed meaning remains documented in section 8.3, while the Appendix wording is reproduced above as supplied. The canonical definition should be verified before external publication.

Additional confirmed webhook codes

The following codes were confirmed separately for the delivery webhook and are retained in this guide in addition to the supplied Appendix table:

Error ID	Confirmed webhook meaning
2035	Sender ID is not authorized on the account. The sender must be approved by Cellact Support before it can be used.
2055	No suitable balance is available for the destination number type. This can occur for a kosher number or an international number when the account does not have the required balance and authorization.
4002	Account balance exhausted (confirmed webhook description; see discrepancy note above).