Lertify API documentation

Date: 1. 7. 2026

The Lertify SMS API allows sending SMS messages to one or multiple recipients via a simple HTTP interface.

Endpoint

```
POST https://api.lertify.app/v1/sms/messages
```

Authentication

Requests must include an API key in the HTTP header.

Header	Type	Required	Description
apikey	string	Yes	API key used for authentication

Request

Content-Type: application/json

Body Parameters

Field	Type	Required	Description
sender	string	Yes	Sender name or phone number displayed to recipient
destinations	array[string]	Yes	List of recipient phone numbers
message.content	string	Yes	SMS message content
message.isUnicode	boolean	No	Indicates whether the message uses Unicode
deliveryReportUrl	string	No	URL where delivery reports will be sent
scheduledTime	long	No	Unix timestamp in seconds specifying the scheduled send time. If in the past, the message is sent immediately

Example Request

```
{
  "sender": "Lertify01",
  "destinations": [
    "+38640123456",
    "+38640987654"
  ],
  "message": {
    "content": "Test SMS message",
    "isUnicode": false
  },
  "deliveryReportUrl": "https://example.com/delivery-report",
  "scheduledTime": 1789297200
}
```

Response

Status code

Code	Description
200	Request accepted for processing

Response structure

Field	Type	Description
messages	array	List of sent messages
messages[].messageId	UUID	Unique message identifier
messages[].destination	string	Recipient phone number
messages[].scheduledTime	long	**Included only if the message is scheduled**. Unix timestamp in seconds
messages[].scheduledAt	datetime	**Included only if the message is scheduled**. Scheduled sending time in UTC.
status	string	Request status

Example response:


```
{
  "messages": [
    {
      "messageId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
      "destination": "+38640123456",

    }
  ],
  "status": "scheduled",
  "scheduledTime": 1789297200,
  "scheduledAt": "2026-09-13T09:00:00Z"
}
```


Notes

- The endpoint supports sending multiple SMS messages in a single request
- Phone numbers must be in international format with a + prefix (e.g. +38640123456)
- deliveryReportUrl enables asynchronous delivery status notifications

Delivery report (Webhook)

Overview

If deliveryReportUrl is provided in the request, the system will send HTTP POST requests to the specified URL with information about SMS delivery status.

If deliveryReportUrl is not provided in the request, **the system will use the deliveryReportUrl configured for the sender**, if one is configured. If neither is configured, no delivery report webhook will be sent.

Endpoint Requirements

- Must accept **HTTP POST** requests
- Must return **HTTP 2xx status**

If a non-2xx response is returned, the retry mechanism is triggered.

Delivery report statuses

The delivery report webhook uses the following statuses:

Status	Description
SENT	The message was successfully handed over to the SMS provider or carrier and is awaiting final delivery
DELIVERED	The message was successfully delivered to the recipient's device
FAILED	Delivery failed or an error occurred during processing or sending
REJECTED	The message was rejected before delivery
UNKNOWN	The final delivery status could not be determined
CANCELLED	The message was cancelled before delivery

Notes

- The **ACCEPTED** status in the initial API response indicates that the request was accepted for processing.
- The **SENT** status indicates that the message was successfully handed over to the SMS provider or carrier.
- The **FAILED** status may represent sending errors, unsuccessful delivery, delivery timeout expiration.
- The **REJECTED** status may represent insufficient balance, rejection by the system or carrier.

Delivery report behavior

Delivery reports may contain multiple status updates for the same message during its lifecycle. Each delivery report webhook contains the current message status for the specific recipient.
Example flow: accepted → SENT → DELIVERED

Inbound (Webhook)

If an inboundUrl is configured for the sender, the system will send an HTTP POST request to that URL whenever an SMS message is received from an end user.

Inbound webhooks are used for messages sent by end users to a phone number or sender configured in the Lertify system.

Endpoint requirements:

- Must accept **HTTP POST** requests
- Must return an **HTTP 2xx** status

If the endpoint does not return an HTTP 2xx response, the retry mechanism is triggered.

Webhook request structure:

Field	Type	Description
Sender	string	Phone number of the user who sent the message
Recipient	string	Sender or phone number to which the message was sent
Message	string	Content of the received SMS message
Time	string	Time the message was received in yyyy-MM-dd HH:mm:ss format

Example webhook request:

```
{
  "Sender": "+38640123456",
  "Recipient": "+38631234567",
  "Message": "TEST",
  "Time": "2026-05-15 14:32:10"
}
```

Notes:

- **Inbound webhooks are sent only if an inboundUrl is configured for the sender.**
- The Time field is sent as a string in the yyyy-MM-dd HH:mm:ss format.

Retry Mechanism

If the endpoint does not return a successful response (HTTP 2xx), the system retries delivery using the following form:

```
delay = (poskus²) + 1 minutes
```

- Maximum number of attempts: 10
- Retry intervals increase progressively

Retry Schedule

Attempt	Delay	Total elapsed time
1	2 min	2 min
2	5 min	7 min
3	10 min	17 min
4	17 min	34 min
5	26 min	60 min (1 h)
6	37 min	97 min (1 h37 min)
7	50 min	147 min (2 h 27 min)
8	65 min (1 h 5 min)	212 min (3 h 32 min)
9	82 min (1 h 22 min)	294 min (4 h 54 min)
10	101 min (1 h 41 min)	395 min (6 h 35 min)

Example webhook payload

```
{
  "messageId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "status": "DELIVERED"
}
```

Notes

- Delivery reports are sent per recipient

Code Examples

PHP

```php
<?php

$url = "https://api.lertify.app/v1/sms/messages";
$apiKey = "YOUR_API_KEY";

$data = [
    "sender" => "Lertify01",
    "destinations" => [
        "+38640123456"
    ],
    "message" => [
        "content" => "Test SMS message",
        "isUnicode" => false
    ],
    "deliveryReportUrl" => "https://example.com/delivery-report"
];

$ch = curl_init($url);

curl_setopt($ch, CURLOPT_HTTPHEADER, [
    "Content-Type: application/json",
    "apikey: " . $apiKey
]);

curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_POST, true);
curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($data));

$response = curl_exec($ch);

if (curl_errno($ch)) {
    echo 'Error:' . curl_error($ch);
} else {
    echo "Response: " . $response;
}

curl_close($ch);
```

Python

```python
import requests

url = "https://api.lertify.app/v1/sms/messages"

headers = {
    "Content-Type": "application/json",
    "apikey": "YOUR_API_KEY"
}

data = {
    "sender": "Lertify01",
    "destinations": [
        "+38640123456",
        "+38640987654"
    ],
    "message": {
        "content": "Test SMS message",
        "isUnicode": False
    },
    "deliveryReportUrl": "https://example.com/delivery-report"
}

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

print("Status code:", response.status_code)
print("Response:", response.json())
```