



# SMS Documentation

## SMS API — Integration Guide

This document provides a production-grade reference for sending SMS/MMS via the API and configuring webhooks.

*"Base URL: `https://api.mysms.ca`"*

### Authentication

---

All API requests must include an API key issued to your account.

- Header: `Authorization: Bearer YOUR_API_KEY`
- You can view and manage your key in the portal (Account Settings).

If the key is missing or invalid, the request is rejected with an error.

### Sending SMS

---

- Method: `POST`
- URL: `<base_url>/send`
- Auth: `Authorization: Bearer YOUR_API_KEY`
- Content-Type: `application/json`

## Request Body

---

```
{
  "from": "+15145551212",
  "to": "+15145550000",
  "text": "Hello from API"
}
```

- **from** (string, required): Your assigned SMS DID in E.164 format (e.g., ``+15145551212`` ).
- **to** (string, required): Destination phone number in E.164 format (e.g., ``+15145550000`` ).
- **text** (string, required): Message text.

## Success Response (200)

---

```
{
  "success": true,
  "code": "200",
  "message": "SMS SENT",
  "message_id": "<provider_msg_id>"
}
```

## cURL

---

```
curl -X POST "<base_url>/send" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "from": "+15145551212",
    "to": "+15145550000",
    "text": "Hello from API"
  }'
```

## PHP (Guzzle)

```
$client = new \GuzzleHttp\Client();
$response = $client->post('<base_url>/send', [
    'headers' => [
        'Authorization' => 'Bearer ' . getenv('API_KEY'),
        'Content-Type' => 'application/json',
    ],
    'json' => [
        'from' => '+15145551212',
        'to' => '+15145550000',
        'text' => 'Hello from API'
    ],
]);
echo $response->getStatusCode();
echo (string) $response->getBody();
```

## Sending MMS

---

- Method: `POST`
- URL: `<base_url>/send`
- Auth: `Authorization: Bearer YOUR_API_KEY`
- Content-Type: `application/json`

Files should be accessible via HTTPS and within supported size/type limits.

## Request Body (MMS)

---

```
{
  "from": "+15145551212",
  "to": "+15145550000",
  "attachments": [
    "https://example.com/image1.jpg",
    "https://example.com/image2.png"
  ]
}
```

## Notes

- ``from`` and ``to`` must be E.164 (e.g., ``+1514555...``).
- Supported content types: ``image/jpeg`` , ``image/jpg`` , ``image/png`` , ``image/gif`` , ``video/mp4`` , ``audio/mp3`` , ``audio/mpeg`` , ``application/pdf`` , ``text/plain`` .

## Response Codes

- ``000`` Invalid API key
- ``001`` DID not assigned
- ``002`` API functionality is not enabled on this DID
- ``200`` SMS sent successfully

---

## Updating Webhook

---

Use this to update your webhook URL.

- Method: ``POST``
- URL: ``<base_url>/webhook``
- Auth: ``Authorization: Bearer YOUR_API_KEY``
- Content-Type: ``application/json``

## Request Body

---

```
{
  "url": "https://yourapp.example.com/webhooks/sms"
}
```

## Response (200)

---

```
{
  "code": "200",
  "message": "WEBHOOK URL UPDATED"
}
```

## Response Codes

- `400` Invalid API key
- `200` WEBHOOK URL UPDATED

---

## Inbound SMS Webhook Payload

---

When an inbound SMS is received on one of your DID's, the system sends a `POST` request to your configured webhook URL with the following JSON payload.

### Payload Structure

```
{
  "data": {
    "event_type": "message.received",
    "id": "<uuid>",
    "completed_at": null,
    "cost": null,
    "direction": "inbound",
    "errors": [],
    "payload": {
      "id": "<uuid>",
      "from": {
        "phone_number": "+15145551212"
      },
      "text": "Hello from the sender",
      "to": [
        {
          "phone_number": "+15145550000"
        }
      ]
    }
  }
}
```

```

    ],
    "received_at": "2026-02-12T15:39:43+00:00",
    "record_type": "message"
  },
  "record_type": "event"
}
}
}

```

## Field Reference

Field	Type	Description
<code>`data.event_type`</code>	string	Always <code>`message.received`</code>
<code>`data.id`</code>	string (UUID)	Unique identifier for the event
<code>`data.direction`</code>	string	Always <code>`inbound`</code>
<code>`data.completed_at`</code>	null	Reserved for future use
<code>`data.cost`</code>	null	Reserved for future use
<code>`data.errors`</code>	array	Empty array, reserved for future use
<code>`data.payload.id`</code>	string (UUID)	Unique identifier for the message
<code>`data.payload.from.phone_number`</code>	string	Sender's number in E.164 format
<code>`data.payload.text`</code>	string	Message body
<code>`data.payload.to[].phone_number`</code>	string	Your DID that received the message, in E.164 format
<code>`data.payload.received_at`</code>	string	ISO 8601 timestamp
<code>`data.payload.record_type`</code>	string	Always <code>`message`</code>
<code>`data.record_type`</code>	string	Always <code>`event`</code>

## Inbound MMS (with Media)

When the inbound message includes media attachments (MMS), a `media` array is added to `data.payload` :

```
{
  "data": {
    "event_type": "message.received",
    "id": "<uuid>",
    "direction": "inbound",
    "errors": [],
    "payload": {
      "id": "<uuid>",
      "from": {
        "phone_number": "+15145551212"
      },
      "text": "",
      "to": [
        {
          "phone_number": "+15145550000"
        }
      ],
      "media": [
        { "url": "https://ww2.mysms.ca/storage/mms/image1.jpg" },
        { "url": "https://ww2.mysms.ca/storage/mms/image2.png" }
      ],
      "received_at": "2026-02-12T15:39:43+00:00",
      "record_type": "message"
    },
    "record_type": "event"
  }
}
```

Field	Type	Description
<code>data.payload.media</code>	array	Present only when the message has attachments
<code>data.payload.media[].url</code>	string	Public URL to the media file

***“Tips:”***

- *Ensure the ``from`` number is assigned to your user.*
- *Ensure the API is enabled for that DID (toggle in the portal).*
- *Ensure your ``Authorization`` header contains a valid API key.*
- *Your webhook endpoint must respond with a ``2xx`` status code to acknowledge receipt.*