

# MessageNow: SMS API Documentation

## Introduction

Message Now is a versatile messaging API designed to facilitate messaging via SMS and RCS.

## Help & Support

For implementation support and any feedback, please contact us at:

[support@messagecentral.com](mailto:support@messagecentral.com)

## Message Sending Modes

Message Central offers two primary modes for sending messages through the Message Now API:

### 1. Sending Messages to Single Numbers

- To send a message to a single number, populate the following API fields:
- **MobileNumber:** The recipient's mobile number.
- **Message:** The text content of the message to be sent.

### 2. Sending Messages to Multiple Numbers

There are two methods to send messages to multiple numbers:

#### A. Using Multipart File Upload

- **Upload File:** Use the (file) API parameter to upload a multipart file.
- **Message:** Use the API parameter (message) for the text to be sent.
- **File Format:** Only .xlsx files are supported.
- **Customization:** This method supports customized messages per user, allowing different message content for each recipient listed in the file.

#### B. Using API Fields for Multiple Numbers

- **PhoneNumbers:** Use the API parameter (PhoneNumbers) to specify a list of recipient numbers. (format: <space>country phone: For example - 91 9816111111, 91 7715131105,...)
- **Message:** Use the API parameter (message) for the text to be sent.
- **File Format:** Only .xlsx files are supported.
- **Customization:** This method supports customized messages per user, allowing different message content for each recipient listed in the file.

Difference Between Multipart File Upload and Phone Number Field:

- **Multipart File Upload (.xlsx):** Allows for customized messages per user. Each recipient can receive a unique message as specified in the uploaded .xlsx file.
- **PhoneNumbers field:** Sends the same text to all recipients. The message specified in the message field will be sent uniformly to everyone listed in the PhoneNumbers field.

Please find sample file format in XLXS format below:

Country	Type	Var 1	Var 2	Var 3
91	12345678901	What	\$35	DISCOUNT20
91	23456789012	Kunal	\$60	DISCOUNT20
91	12345678901	Prakash	\$35	DISCOUNT20

Where 'var1', 'var2', and 'var3' are placeholders for variables.

If a company wants to send a custom message using file upload, the message field should be formatted as follows:

Example message - “Hello # #var ##, your balance is # #var2 ## on date # #var3 ##

## Generating Tokens

Request Header	Type	Mandatory?
authToken	String	Yes

To send an SMS to a mobile number, the following request parameters are required. An authentication token, generated by the token generation API, is required to send the SMS.

```
https://cpaas.messagecentral.com
```

When using the Message Now SMS verification API to send SMS, the initial call must be to the token generation API.

# MessageNow: SMS API Documentation

This API returns a token that must be included in all subsequent calls. The authentication token is required to validate the user and must be included in the header of every request.

Field	Type	Mandatory?	Description
customerId	String	yes	Customer identifier (need to signup to get your customer id)
country	String	no	Country code to send OTP to
email	String	no	Email
key	String	yes	Base-64 encrypted password
scope	String	no	Use 'NEW' for first time


## Request URL Path:

```
/auth/v1/authentication/token
```

## cURL

All Platform API endpoints below must begin with the following URL:

```
1 curl --location 'https://cpaas.messagecentral.com/auth/v1/  
2 authentication/token?  
3 customerId=%3CCustomerId%3E&key=%3CBase64%20Encrypted%20password%3E  
4 &scope=NEW&country=91&email=test%40messagecentral.com' \  
5 --header 'accept: */*'
```

 **NOTE:** To convert a cURL command into code using Postman, open Postman, import the cURL command via the "Import" button, and then generate the code in your preferred language by clicking the "Code" button on the right side of the request.

# MessageNow: SMS API Documentation

## JSON Response

A successful response will return a status code of 200.

```
1  {
2    "responseCode": 200,
3    "message": "SUCCESS",
4    "data": {
5      "verificationId": "xxxx",
6      "mobileNumber": "xxxx",
7      "responseCode": "200",
8      "errorMessage": ` ,
9      "verificationStatus": "VERIFICATION_COMPLETED",
10     "authToken": null,
11     "transactionId": "xxxx"
12   }
13 }
```

## Response Code

Code	Display Text
200	SUCCESS
400	BAD_REQUEST
409	DUPLICATE_RESOURCE
500	SERVER_ERROR
501	INVALID_CUSTOMER_ID
505	INVALID_VERIFICATION_ID
506	REQUEST_ALREADY_EXISTS
511	INVALID_COUNTRY_CODE
700	VERIFICATION_FAILED
702	WRONG_OTP_PROVIDED
703	ALREADY_VERIFIED
705	VERIFICATION_EXPIRED
800	MAXIMUM_LIMIT_REACHED

# MessageNow: SMS API Documentation

## Send SMS

To send SMS on a mobile number below are the request parameters. The authentication token is required to send SMS which is generated by the generated token API (which you can find above in Introduction section).

### Request URL Path:

A successful response will return a status code of 200.


```
/verification/v3/send
```

### Request Parameters:

A successful response will return a status code of 200.


Field	Type	Mandatory?	Description
type	String	yes	Use "SMS" while sending SMS
file	xlsx	no	This needs to be sent as form data
countryCode	String	Yes (in case of single SMS & Verify Now)	CountryCode
mobileNumber	String	Yes (in case of single SMS)	Mobile number for single text
flowType	String	Yes	Send "SMS"
messageType	String	No (default is OTP)	The type of message to be sent (Allowed values are TRANSACTION, PROMOTIONAL, OTP)

senderId	String	Yes (in case of SMS)	ID of the sender
phoneNumbers	String	No	A list of phone numbers to send the SMS (format: country <space>mobile : 91 9876111111)
message	String	No	The custom message to be sent
templateId	String	No	Template Id of your approved template
entityId	String	No	Entity Id of your brand

 **NOTE:** TemplateID and EntityId are required if the template is not registered with the Message Center. Also, TemplateID and EntityId must be present or both must be missing.

**cURL**

```
curl --location --request POST 'https://cpaas.messagecentral.com/  
verification/v3/send?countryCode=91&flowT  
ype=SMS&mobileNumber=9999999999&senderId=UTOMOB&type=SMS&message=%3CYo  
ur%20 Message%20Template%3E&messageType=PROMOTIONAL '  
\--header 'authToken:  
eyJhbGciOiJIUzUxMiJ9.eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiIsInR5cGEiOiJ0bm90cmVudCI6dXNpdDQ4GdEjM5WntEGCw0hITng'  
Z2kTI_Mg_EAaKVvCcDpW_TYzyzdTd94GdEjM5WntEGCw0hITng'
```

 **NOTE:** To convert cURL commands to code using Postman, open Postman, import the cURL commands via the “Import” button, then generate the code in your preferred language by clicking the “Code” button on the right side of the request.

A successful response will return a status code of 200.

## Code Example

## JAVA

```
OkHttpClient client = new OkHttpClient().newBuilder()
    .build();
MediaType mediaType = MediaType.parse("text/plain");
RequestBody body = RequestBody.create(mediaType, "");
Request request = new Request.Builder()
    .url("https://cpaas.messagecentral.com/auth/v1/authentication/token?
customerId=<CustomerId>&key=<Base64 Encrypted
password>&scope=NEW&country=91&email=test@messagecentral.com")
    .method("GET", body)
    .addHeader("accept", "*/*")
    .build();
Response response = client.newCall(request).execute();
```

## Node.js

```
var request = require('request');
var options = {
  'method': 'GET',
  'url': 'https://cpaas.messagecentral.com/auth/v1/authentication/
token?customerId=<CustomerId>&key=<Base64 Encrypted
password>&scope=NEW&country=91&email=test@messagecentral.com',
  'headers': {
    'accept': '*/*'
  }
};
request(options, function (error, response) {
  if (error) throw new Error(error);
  console.log(response.body);
});
```



## PHP

```
1  <?php
2  require_once 'HTTP/Request2.php';
3  $request = new HTTP_Request2();
4  $request->setUrl('https://cpaas.messagecentral.com/auth/v1/
5  authentication/token?customerId=<CustomerId>&key=<Base64 Encrypted
6  password>&scope=NEW&country=91&email=test@messagecentral.com');
7  $request->setMethod(HTTP_Request2::METHOD_GET);
8  $request->setConfig(array(
9      'follow_redirects' => TRUE
10 ));
11 $request->setHeader(array(
12     'accept' => '*/*'
13 ));
14 try {
15     $response = $request->send();
16     if ($response->getStatus() == 200) {
17         echo $response->getBody();
18     }
19     else {
20         echo 'Unexpected HTTP status: ' . $response->getStatus() . ' '
21         .
22         $response->getReasonPhrase();
23     }
24 }
25 catch(HTTP_Request2_Exception $e) {
26     echo 'Error: ' . $e->getMessage();
27 }
```

## Python

```
import requests

url = "https://cpaas.messagecentral.com/auth/v1/authentication/token?
customerId=<CustomerId>&key=<Base64 Encrypted
password>&scope=NEW&country=91&email=test@messagecentral.com"

payload = {}
headers = {
    'accept': '*/*'
}

response = requests.request("GET", url, headers=headers, data=payload)

print(response.text)
```

## Ruby

```
1 require "uri"
2 require "net/http"
3
4 url = URI("https://cpaas.messagecentral.com/auth/v1/
5 authentication/token?customerId=<CustomerId>&key=<Base64 Encrypted
6 password>&scope=NEW&country=91&email=test@messagecentral.com")
7
8 https = Net::HTTP.new(url.host, url.port)
9 https.use_ssl = true
10
11 request = Net::HTTP::Get.new(url)
12 request["accept"] = "*/*"
13
14 response = https.request(request)
15 puts response.read_body
```

## C#

```
1 var client = new HttpClient();
2 var request = new HttpRequestMessage(HttpMethod.Get, "https://
3 cpaas.messagecentral.com/auth/v1/authentication/token?
4 customerId=<CustomerId>&key=<Base64 Encrypted
5 password>&scope=NEW&country=91&email=test@messagecentral.com");
6 request.Headers.Add("accept", "*/*");
7 var response = await client.SendAsync(request);
8 response.EnsureSuccessStatusCode();
9 Console.WriteLine(await response.Content.ReadAsStringAsync());
10 Send OTP
```