

SMSGO SMS Gateway API

Version 1.8.0

<https://www.smsgo.com.tw>

Contact: [SMSGO Support](#)

Overview

Send SMS messages through the SMSGO platform via HTTP POST.

Key Features:

- **Multiple Encodings:** BIG5 (Traditional Chinese), ASCII, UCS2 (Unicode).
 - **Long SMS:** Up to 335 characters.
 - **WAP Push:** Send links to mobile devices.
 - **Scheduled Delivery:** Schedule future sends.
 - **Callbacks:** Webhooks for delivery status and user replies.
-

Notes:

1. BIG5/ASCII/UCS2 support single SMS only. Messages exceeding the limit are auto-truncated.
 2. PBIG5/PASCII are POPUP. LBIG5/LASCII/LUCS2 are long SMS (max 335 Chinese chars).
 3. For PUSH, smbody+wapurl combined must be ≤ 88 bytes (Chinese chars count as 3 bytes).
 4. Setting replyurl enables SMS reply at +1 point per 10 messages (max URL length: 300 chars).
 5. International SMS: only BIG5/ASCII/UCS2. Use UCS2 for Simplified Chinese.
- > ■■ **Important:** To use this API, you will need to contact customer support to enable it.
-

Contact:

- ■ LINE: [@cdn4573o](#)

Table of Contents

- SMS
- Query
- Account
- Verification

SMS

Operations for sending and managing SMS messages.

GET `/sms_gw/sendsms.aspx`

Send SMS (GET) ■ Deprecated

■ DEPRECATED

■ **Deprecated** — Use POST instead. Credentials appear in server logs.

Send SMS via GET query parameters. All values must be URL-encoded.

Parameters

Name	In	Type	Required	Description
username	query	string	Yes	Member account username
password	query	string	Yes	Member password or API Key
dstaddr	query	string	Yes	Recipient mobile number(s). Multiple numbers comma-separated (max 50). Domestic: 09xxxxxxx. International: prefix with + URL-encoded as %2b, e.g. %2b8613681912700.
smbody	query	string	Yes	SMS content. Max 70 chars (zh/en mix) or 160 chars (ASCII). Long SMS (L* encodings): max 335 chars. WAP Push (PUSH): this is the push title; smbody+wapur1 ≤ 88 bytes.
encoding	query	string (enum)	No	Message encoding. Default: BIG5. Value Description --- --- BIG5 Traditional Chinese, single SMS, max 70 chars ASCII English only, single SMS, max 160 chars UCS2 Unicode (Japanese, Korean, Simplified Chinese) PBIG5 POPUP Traditional Chinese PASCII POPUP English LBIG5 Long SMS Traditional Chinese, max 335 chars LASCII Long SMS English LUCS2 Long SMS Unicode PUSH WAP Push (requires wapur1)
dlvtime	query	string	No	Scheduled delivery time. Format: YYYY/MM/DD hh24:mm:ss. Leave blank to send immediately.
wapur1	query	string	No	WAP Push destination URL. Only used when encoding=PUSH. Combined with smbody must be ≤ 88 bytes.
replyurl	query	string	No	Callback URL to receive user reply messages (extra charge: +1 point per 10 messages). Only for BIG5/ASCII/UCS2. Max 300 characters.
replydays	query	integer	No	Number of days to collect user replies. Default: 3, Max: 30. Only for BIG5/ASCII/UCS2.

response	query	string	No	Callback URL for delivery status reports. System POSTs status updates to this URL. Leave blank to disable.
rtype	query	string (enum)	No	Response format. Omit for plain text, JSON for JSON, XML for XML.

Responses

200

API Send SMS response. Format determined by rtype parameter. statuscode=0 means success.

Error Codes:

Code	Description
0	Success
-1	CGI string error
-2	Username, password, or IP error
-3	Username or password empty
-5	Phone number format error
-8	Insufficient points
-9	Too many recipients (max 50)
-10	Send failed
-11	Database error
-12	Message body empty
-13	Parameter count mismatch
-14	Fraud keyword detected
-15	IP address not allowed
-16	API service not enabled
-17	Callback URL not allowed
-18	Off-net number
-19	Number blocked
-20	Marked as SPAM
-21	Capacity reached
-22	Request rerouted
-23	No signature
-24	OTP verification failed
-25	OTP expired
-30	Operator system ID error

text/plain — Success

```
msgid=100101121234567
statuscode=0
statusstr=OK
point=1
```

text/plain — Error

```
msgid=-3
statuscode=-3
```

```
statusstr=username or password empty!  
point=0
```

application/json — Success

```
{  
  "result": {  
    "msgid": "1504280910499665",  
    "statusCode": "0",  
    "statusstr": "OK",  
    "point": 1  
  }  
}
```

application/json — Error

```
{  
  "result": {  
    "msgid": "-3",  
    "statusCode": "-3",  
    "statusstr": "username or password empty!",  
    "point": 0  
  }  
}
```

application/xml — Success

```
<response><code>0</code><msgid>1688888</msgid><text>Success</text></response>
```

application/xml — Error

```
<response><code>-1</code><msgid></msgid><text>Username or Password Empty</text></response>
```

Code Samples

cURL

```
curl "https://www.smsgo.com.tw/sms_gw/sendsms.aspx?username=your_username&password=your_password&dstaddr=0912345678&smbdy=Test"
```

JavaScript

```
fetch('https://www.smsgo.com.tw/sms_gw/sendsms.aspx?username=your_username&password=your_password&dstaddr=0912345678&smbdy=Test', {  
  method: 'GET'  
})  
.then(response => response.text())  
.then(result => console.log(result))  
.catch(error => console.log('error', error));
```

C#

```
using System;  
using System.Net.Http;  
using System.Threading.Tasks;  
  
class Program  
{  
    static async Task Main(string[] args)  
    {  
        var client = new HttpClient();  
        var request = new HttpRequestMessage(HttpMethod.Get, "https://www.smsgo.com.tw/sms_gw/sendsms.aspx?username=your_username&password=your_password&dstaddr=0912345678&smbdy=Test");  
        var response = await client.SendAsync(request);  
        response.EnsureSuccessStatusCode();  
    }  
}
```

```

        Console.WriteLine(await response.Content.ReadAsStringAsync());
    }
}

```

PHP

```

<?php

$url = 'https://www.smsgo.com.tw/sms_gw/sendsms.aspx?username=your_username&password=your_password&dstaddr=0912345678&smbody=Test';
$result = file_get_contents($url);

if ($result === FALSE) {
    /* Handle error */
    echo "Error sending request.";
}

var_dump($result);

?>

```

Python

```

import requests

url = 'https://www.smsgo.com.tw/sms_gw/sendsms.aspx?username=your_username&password=your_password&dstaddr=0912345678&smbody=Test'

try:
    response = requests.get(url)
    response.raise_for_status()
    print("Response from server:")
    print(response.text)
except requests.exceptions.RequestException as e:
    print(f"An error occurred: {e}")

```

Java

```

import java.net.URI;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;

public class SmsGoExample {
    public static void main(String[] args) throws Exception {
        HttpClient client = HttpClient.newHttpClient();
        HttpRequest request = HttpRequest.newBuilder()
            .uri(URI.create("https://www.smsgo.com.tw/sms_gw/sendsms.aspx?username=your_username&password=your_password&dstaddr=0912345678&smbody=Test"))
            .GET()
            .build();

        HttpResponse<String> response = client.send(request, HttpResponse.BodyHandlers.ofString());
        System.out.println(response.body());
    }
}

```

POST

/sms_gw/sendsms.aspx

Send SMS (POST)

Send an SMS message. Accepts `application/x-www-form-urlencoded` or `application/json`.

Request Body

`application/x-www-form-urlencoded`

`application/json`

Responses

200

API Send SMS response. Format determined by `rtype` parameter. `statuscode=0` means success.

Error Codes:

Code	Description
0	Success
-1	CGI string error
-2	Username, password, or IP error
-3	Username or password empty
-5	Phone number format error
-8	Insufficient points
-9	Too many recipients (max 50)
-10	Send failed
-11	Database error
-12	Message body empty
-13	Parameter count mismatch
-14	Fraud keyword detected
-15	IP address not allowed
-16	API service not enabled
-17	Callback URL not allowed
-18	Off-net number
-19	Number blocked
-20	Marked as SPAM
-21	Capacity reached
-22	Request rerouted
-23	No signature
-24	OTP verification failed
-25	OTP expired
-30	Operator system ID error

text/plain — Success

```
msgid=100101121234567
statuscode=0
statusstr=OK
point=1
```

text/plain — Error

```
msgid=-3
statuscode=-3
statusstr=username or password empty!
point=0
```

application/json — Success

```
{
  "result": {
    "msgid": "1504280910499665",
    "statuscode": "0",
    "statusstr": "OK",
    "point": 1
  }
}
```

application/json — Error

```
{
  "result": {
    "msgid": "-3",
    "statuscode": "-3",
    "statusstr": "username or password empty!",
    "point": 0
  }
}
```

application/xml — Success

```
<response><code>0</code><msgid>1688888</msgid><text>Success</text></response>
```

application/xml — Error

```
<response><code>-1</code><msgid></msgid><text>Username or Password Empty</text></response>
```

Code Samples

cURL

```
curl -X POST -d 'username=your_username&password=your_password&dstaddr=0912345678&smbdy=Test'
https://www.smsgo.com.tw/sms_gw/sendsms.aspx
```

JavaScript

```
const myHeaders = new Headers();
myHeaders.append("Content-Type", "application/x-www-form-urlencoded");

const urlencoded = new URLSearchParams();
urlencoded.append("username", "your_username");
urlencoded.append("password", "your_password");
urlencoded.append("dstaddr", "0912345678");
urlencoded.append("smbdy", "Test");

const requestOptions = {
  method: 'POST',
  headers: myHeaders,
  body: urlencoded,
  redirect: 'follow'
};

fetch("https://www.smsgo.com.tw/sms_gw/sendsms.aspx", requestOptions)
  .then(response => response.text())
  .then(result => console.log(result))
  .catch(error => console.log('error', error));
```

C#

```
using System;
using System.Collections.Generic;
using System.Net.Http;
using System.Threading.Tasks;

public class SmsGoExample
{
    public static async Task Main(string[] args)
    {
        var client = new HttpClient();
        var values = new Dictionary<string, string>
        {
            { "username", "your_username" },
            { "password", "your_password" },
            { "dstaddr", "0912345678" },
            { "smbody", "This is a test message from C#." }
        };

        var content = new FormUrlEncodedContent(values);
        var response = await client.PostAsync("https://www.smsgo.com.tw/sms_gw/sendsms.aspx",
content);
        var responseString = await response.Content.ReadAsStringAsync();

        Console.WriteLine(responseString);
    }
}
```

PHP

```
<?php

$url = 'https://www.smsgo.com.tw/sms_gw/sendsms.aspx';
$data = array(
    'username' => 'your_username',
    'password' => 'your_password',
    'dstaddr' => '0912345678',
    'smbody' => 'This is a test message from PHP.'
);

$options = array(
    'http' => array(
        'header' => "Content-type: application/x-www-form-urlencoded\r\n",
        'method' => 'POST',
        'content' => http_build_query($data)
    )
);

$context = stream_context_create($options);
$result = file_get_contents($url, false, $context);

if ($result === FALSE) {
    /* Handle error */
    echo "Error sending request.";
}

var_dump($result);

?>
```

Python

```

import requests

url = 'https://www.smsgo.com.tw/sms_gw/sendsms.aspx'
payload = {
    'username': 'your_username',
    'password': 'your_password',
    'dstaddr': '0912345678',
    'smbody': 'This is a test message from Python.'
}

try:
    response = requests.post(url, data=payload)
    response.raise_for_status() # Raise an exception for bad status codes
    print("Response from server:")
    print(response.text)
except requests.exceptions.RequestException as e:
    print(f"An error occurred: {e}")

```

Java

```

import java.net.URI;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;

public class SmsGoExample {
    public static void main(String[] args) throws Exception {
        HttpClient client = HttpClient.newHttpClient();
        HttpRequest request = HttpRequest.newBuilder()
            .uri(URI.create("https://www.smsgo.com.tw/sms_gw/sendsms.aspx"))
            .header("Content-Type", "application/x-www-form-urlencoded")
            .POST(HttpRequest.BodyPublishers.ofString("username=your_username&password=your_password&dstaddr=0912345678&smbody=Test"))
            .build();

        HttpResponse<String> response = client.send(request, HttpResponse.BodyHandlers.ofString());
        System.out.println(response.body());
    }
}

```

GET /sms_gw/sendsmsCancel.aspx

Cancel Scheduled SMS (GET) ■ Deprecated

■ **DEPRECATED**

■ **Deprecated** — Use POST instead.

Cancel a previously scheduled SMS that has not yet been sent.

Parameters

Name	In	Type	Required	Description
username	query	string	Yes	Member account username
password	query	string	Yes	Member password or API Key
msgid	query	string	Yes	Message ID of the scheduled SMS to cancel.

r type	query	string (enum)	No	Response format. Omit for plain text, JSON for JSON, XML for XML.
--------	-------	---------------	----	---

Responses

200

Cancellation result. 0 = success.

text/plain — Success

```
0
```

text/plain — Error

```
-10
```

application/xml

```
<response><code>0</code><text>Success</text></response>
```

Code Samples

cURL

```
curl "https://www.smsgo.com.tw/sms_gw/sendsmsCancel.aspx?username=your_username&password=your_password&msgid=12345"
```

JavaScript

```
fetch('https://www.smsgo.com.tw/sms_gw/sendsmsCancel.aspx?username=your_username&password=your_password&msgid=12345', {
  method: 'GET'
})
.then(response => response.text())
.then(result => console.log(result))
.catch(error => console.log('error', error));
```

C#

```
using System;
using System.Net.Http;
using System.Threading.Tasks;

class Program
{
    static async Task Main(string[] args)
    {
        var client = new HttpClient();
        var request = new HttpRequestMessage(HttpMethod.Get, "https://www.smsgo.com.tw/sms_gw/sendsmsCancel.aspx?username=your_username&password=your_password&msgid=12345");
        var response = await client.SendAsync(request);
        response.EnsureSuccessStatusCode();
        Console.WriteLine(await response.Content.ReadAsStringAsync());
    }
}
```

PHP

```
<?php

$url = 'https://www.smsgo.com.tw/sms_gw/sendsmsCancel.aspx?username=your_username&password=your_password&msgid=12345';
$result = file_get_contents($url);

if ($result === FALSE) {
```

```

        echo "Error sending request.";
    }

    var_dump($result);

?>

```

Python

```

import requests

url = 'https://www.smsgo.com.tw/sms_gw/sendsmsCancel.aspx?username=your_username&password=your_password&msgid=12345'

try:
    response = requests.get(url)
    response.raise_for_status()
    print(response.text)
except requests.exceptions.RequestException as e:
    print(f"An error occurred: {e}")

```

Java

```

import java.net.URI;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;

public class SmsGoExample {
    public static void main(String[] args) throws Exception {
        HttpClient client = HttpClient.newHttpClient();
        HttpRequest request = HttpRequest.newBuilder()
            .uri(URI.create("https://www.smsgo.com.tw/sms_gw/sendsmsCancel.aspx?username=your_username&password=your_password&msgid=12345"))
            .GET()
            .build();
        HttpResponse<String> response = client.send(request, HttpResponse.BodyHandlers.ofString());
        System.out.println(response.body());
    }
}

```

POST

/sms_gw/sendsmsCancel.aspx

Cancel Scheduled SMS (POST)

Cancel a previously scheduled SMS that has not yet been sent. Accepts application/x-www-form-urlencoded or application/json.

Request Body

application/x-www-form-urlencoded

application/json

Responses

200

Cancellation result. 0 = success.

text/plain

```
0
```

application/xml

```
<response><code>0</code><text>Success</text></response>
```

Code Samples

cURL

```
curl -X POST -d 'username=your_username&password=your_password&msgid=12345' https://www.smsgo.com.tw/sms_gw/sendsmsCancel.aspx
```

JavaScript

```
const myHeaders = new Headers();
myHeaders.append("Content-Type", "application/x-www-form-urlencoded");

const urlencoded = new URLSearchParams();
urlencoded.append("username", "your_username");
urlencoded.append("password", "your_password");
urlencoded.append("msgid", "12345");

const requestOptions = {
  method: 'POST',
  headers: myHeaders,
  body: urlencoded,
  redirect: 'follow'
};

fetch("https://www.smsgo.com.tw/sms_gw/sendsmsCancel.aspx", requestOptions)
  .then(response => response.text())
  .then(result => console.log(result))
  .catch(error => console.log('error', error));
```

C#

```
using System;
using System.Collections.Generic;
using System.Net.Http;
using System.Threading.Tasks;

public class SmsGoExample
{
    public static async Task Main(string[] args)
    {
        var client = new HttpClient();
        var values = new Dictionary<string, string>
        {
            { "username", "your_username" },
            { "password", "your_password" },
            { "msgid", "12345" }
        };

        var content = new FormUrlEncodedContent(values);
        var response = await client.PostAsync("https://www.smsgo.com.tw/sms_gw/sendsmsCancel.aspx", content);
        var responseString = await response.Content.ReadAsStringAsync();

        Console.WriteLine(responseString);
    }
}
```

```
}
```

PHP

```
<?php

$url = 'https://www.smsgo.com.tw/sms_gw/sendsmsCancel.aspx';
$data = array(
    'username' => 'your_username',
    'password' => 'your_password',
    'msgid' => '12345'
);

$options = array(
    'http' => array(
        'header' => "Content-type: application/x-www-form-urlencoded\r\n",
        'method' => 'POST',
        'content' => http_build_query($data)
    )
);
$context = stream_context_create($options);
$result = file_get_contents($url, false, $context);

if ($result === FALSE) {
    echo "Error sending request.";
}

var_dump($result);

?>
```

Python

```
import requests

url = 'https://www.smsgo.com.tw/sms_gw/sendsmsCancel.aspx'
payload = {
    'username': 'your_username',
    'password': 'your_password',
    'msgid': '12345'
}

try:
    response = requests.post(url, data=payload)
    response.raise_for_status()
    print(response.text)
except requests.exceptions.RequestException as e:
    print(f"An error occurred: {e}")
```

Java

```
import java.net.URI;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;

public class SmsGoExample {
    public static void main(String[] args) throws Exception {
        HttpClient client = HttpClient.newHttpClient();
        HttpRequest request = HttpRequest.newBuilder()
            .uri(URI.create("https://www.smsgo.com.tw/sms_gw/sendsmsCancel.aspx"))
            .header("Content-Type", "application/x-www-form-urlencoded")
            .POST(HttpRequest.BodyPublishers.ofString("username=your_username&password=you
```

```

r_password&msgid=12345"))
    .build();
    HttpResponse<String> response = client.send(request, HttpResponse.BodyHandlers.ofString());
    System.out.println(response.body());
}
}

```

POST

/sms_gw/sendsmscustom.aspx

Send Personalized Bulk SMS

Send a batch of SMS messages where each recipient receives a unique message body.

Accepts three input modes (auto-detected from Content-Type):

Mode	Content-Type	How to pass data
JSON array	application/json	messages array of {dstaddr, smbody} objects
JSON CSV	application/json	csv string field
Form CSV	application/x-www-form-urlencoded	csv string field
File upload	multipart/form-data	csv file field (UTF-8 or BIG5/CP950)

CSV format: one mobile, message per line. Only the first comma splits the columns, so commas in the message are safe. Lines starting with # are ignored.

> ■■ This endpoint is intended for internal/MCP use and does not require IP whitelisting.

Request Body

application/json

application/x-www-form-urlencoded

multipart/form-data

Responses

200

Batch send result. statuscode=0 means all sent. statuscode=1 means partial. statuscode=-10 means all failed.

application/json — All sent

```

{
  "statusCode": "0",
  "statusstr": "OK",
  "total": 2,
  "sent": 2,
  "failed": 0,
  "results": [
    {
      "seq": 1,
      "dstaddr": "0912345678",
      "statusCode": "0",
      "statusstr": "OK",

```

```

"msgid": "S202605281200000001",
"point": 998
},
{
"seq": 2,
"dstaddr": "0987654321",
"statuscode": "0",
"statusstr": "OK",
"msgid": "S202605281200000002",
"point": 997
}
]
}

```

application/json — Partial success

```

{
"statuscode": "1",
"statusstr": "partial",
"total": 2,
"sent": 1,
"failed": 1,
"results": [
{
"seq": 1,
"dstaddr": "0912345678",
"statuscode": "0",
"statusstr": "OK",
"msgid": "S202605281200000001",
"point": 998
},
{
"seq": 2,
"dstaddr": "invalid",
"statuscode": "-5",
"statusstr": "Phone number format error",
"msgid": "",
"point": 998
}
]
}

```

Code Samples

cURL (JSON array)

```

curl -X POST https://www.smsgo.com.tw/sms_gw/sendsmscustom.aspx \
-H 'Content-Type: application/json' \
-d '{"username":"your_username","password":"your_password","messages":[{"dstaddr":"0912345678","smbdy":"Hello Alice"}, {"dstaddr":"0987654321","smbdy":"Hello Bob"}]}'

```

cURL (file upload)

```

curl -X POST https://www.smsgo.com.tw/sms_gw/sendsmscustom.aspx \
-F 'username=your_username' \
-F 'password=your_password' \
-F 'csv=@/path/to/messages.csv'

```

Python (JSON)

```

import requests

url = 'https://www.smsgo.com.tw/sms_gw/sendsmscustom.aspx'
payload = {

```

```
'username': 'your_username',
'password': 'your_password',
'messages': [
    {'dstaddr': '0912345678', 'smbody': 'Hello Alice'},
    {'dstaddr': '0987654321', 'smbody': 'Hello Bob'}
]
}
response = requests.post(url, json=payload)
print(response.json())
```

Python (file upload)

```
import requests

url = 'https://www.smsgo.com.tw/sms_gw/sendsmscustom.aspx'
with open('messages.csv', 'rb') as f:
    response = requests.post(url, data={
        'username': 'your_username',
        'password': 'your_password'
    }, files={'csv': f})
print(response.json())
```

Query

Operations for querying message delivery status.

GET /sms_gw/query.aspx

Query SMS Status (GET) ■ Deprecated

■ **DEPRECATED**

■ **Deprecated** — Use POST instead.

Query delivery status of a single SMS.

Parameters

Name	In	Type	Required	Description
username	query	string	Yes	Member account username
password	query	string	Yes	Member password or API Key
msgid	query	string	Yes	Message ID returned from a successful send request.
dstaddr	query	string	Yes	Recipient mobile number(s). Multiple numbers comma-separated (max 50). Domestic: 09xxxxxxx. International: prefix with + URL-encoded as %2b, e.g. %2b8613681912700.
rtype	query	string (enum)	No	Response format. Omit for plain text, JSON for JSON, XML for XML.

Responses

200

Delivery status. Plain text format: statusCode|statusStr|cellphone

statusStr values: DELIVRD, REJECTD, EXPIRED, UNDELIV, UNKNOWN, ENROUTE

text/plain — Delivered

```
0|DELIVRD|0912345678
```

text/plain — Serial number not found

```
-2|-2|can not find this serial number!
```

application/json — Delivered

```
{
  "results": {
    "result": [
      {
        "statusCode": "0",
        "statusStr": "DELIVRD",
        "cellphone": "0912345678"
      }
    ]
  }
}
```

application/json — Serial number not found

```
{
  "result": {
    "msgid": "-1",
    "statusCode": "-1",
    "statusstr": "CGI string error!!",
    "point": 0
  }
}
```

application/xml — Success

```
<results><result><statusCode>0</statusCode><statusStr>DELIVRD</statusStr><cellphone>0912345678</cellphone></result></results>
```

application/xml — Serial number not found

```
<results><r><statusCode>-2</statusCode><statusStr>can not find this serial number!</statusStr></r></results>
```

Code Samples

cURL

```
curl "https://www.smsgo.com.tw/sms_gw/query.aspx?username=your_username&password=your_password&msgid=12345&dstaddr=0912345678"
```

JavaScript

```
fetch('https://www.smsgo.com.tw/sms_gw/query.aspx?username=your_username&password=your_password&msgid=12345&dstaddr=0912345678', {
  method: 'GET'
})
.then(response => response.text())
.then(result => console.log(result))
.catch(error => console.log('error', error));
```

C#

```
using System;
using System.Net.Http;
using System.Threading.Tasks;

class Program
{
    static async Task Main(string[] args)
    {
        var client = new HttpClient();
        var request = new HttpRequestMessage(HttpMethod.Get, "https://www.smsgo.com.tw/sms_gw/query.aspx?username=your_username&password=your_password&msgid=12345&dstaddr=0912345678");
        var response = await client.SendAsync(request);
        response.EnsureSuccessStatusCode();
        Console.WriteLine(await response.Content.ReadAsStringAsync());
    }
}
```

PHP

```
<?php

$url = 'https://www.smsgo.com.tw/sms_gw/query.aspx?username=your_username&password=your_password&msgid=12345&dstaddr=0912345678';
$result = file_get_contents($url);

if ($result === FALSE) {
    /* Handle error */
    echo "Error sending request.";
}
```

```
}  
  
var_dump($result);  
  
?>
```

Python

```
import requests  
  
url = 'https://www.smsgo.com.tw/sms_gw/query.aspx?username=your_username&password=your_password&msgid=12345&dstaddr=0912345678'  
  
try:  
    response = requests.get(url)  
    response.raise_for_status()  
    print("Response from server:")  
    print(response.text)  
except requests.exceptions.RequestException as e:  
    print(f"An error occurred: {e}")
```

Java

```
import java.net.URI;  
import java.net.http.HttpClient;  
import java.net.http.HttpRequest;  
import java.net.http.HttpResponse;  
  
public class SmsGoExample {  
    public static void main(String[] args) throws Exception {  
        HttpClient client = HttpClient.newHttpClient();  
        HttpRequest request = HttpRequest.newBuilder()  
            .uri(URI.create("https://www.smsgo.com.tw/sms_gw/query.aspx?username=your_username&password=your_password&msgid=12345&dstaddr=0912345678"))  
            .GET()  
            .build();  
  
        HttpResponse<String> response = client.send(request, HttpResponse.BodyHandlers.ofString());  
        System.out.println(response.body());  
    }  
}
```

POST

/sms_gw/query.aspx

Query SMS Status (POST)

Query delivery status of a single SMS via POST. Accepts application/x-www-form-urlencoded or application/json.

Request Body

application/x-www-form-urlencoded

application/json

Responses

200

Delivery status. Plain text: statusCode|statusStr|cellphone

statusStr values: DELIVRD, REJECTD, EXPIRED, UNDELIV, UNKNOWN, ENROUTE

text/plain — Delivered

```
0|DELIVRD|0912345678
```

text/plain — Serial number not found

```
-2|-2|can not find this serial number!
```

application/json — Delivered

```
{
  "results": {
    "result": [
      {
        "statusCode": "0",
        "statusStr": "DELIVRD",
        "cellphone": "0912345678"
      }
    ]
  }
}
```

application/json — Serial number not found

```
{
  "result": {
    "msgid": "-1",
    "statusCode": "-1",
    "statusStr": "CGI string error!!",
    "point": 0
  }
}
```

application/xml — Success

```
<results><result><statusCode>0</statusCode><statusStr>DELIVRD</statusStr><cellphone>0912345678</cellphone></result></results>
```

application/xml — Serial number not found

```
<results><r><statusCode>-2</statusCode><statusStr>can not find this serial number!</statusStr></r></results>
```

Code Samples

cURL

```
curl -X POST -d 'username=your_username&password=your_password&msgid=12345&dstaddr=0912345678' https://www.smsgo.com.tw/sms_gw/query.aspx
```

JavaScript

```
const myHeaders = new Headers();
myHeaders.append("Content-Type", "application/x-www-form-urlencoded");

const urlencoded = new URLSearchParams();
urlencoded.append("username", "your_username");
urlencoded.append("password", "your_password");
urlencoded.append("msgid", "12345");
urlencoded.append("dstaddr", "0912345678");

const requestOptions = {
  method: 'POST',
```

```

    headers: myHeaders,
    body: urlencoded,
    redirect: 'follow'
  };

  fetch("https://www.smsgo.com.tw/sms_gw/query.aspx", requestOptions)
    .then(response => response.text())
    .then(result => console.log(result))
    .catch(error => console.log('error', error));

```

C#

```

using System;
using System.Collections.Generic;
using System.Net.Http;
using System.Threading.Tasks;

public class SmsGoExample
{
    public static async Task Main(string[] args)
    {
        var client = new HttpClient();
        var values = new Dictionary<string, string>
        {
            { "username", "your_username" },
            { "password", "your_password" },
            { "msgid", "12345" },
            { "dstaddr", "0912345678" }
        };

        var content = new FormUrlEncodedContent(values);
        var response = await client.PostAsync("https://www.smsgo.com.tw/sms_gw/query.aspx", content);
        var responseString = await response.Content.ReadAsStringAsync();

        Console.WriteLine(responseString);
    }
}

```

PHP

```

<?php

$url = 'https://www.smsgo.com.tw/sms_gw/query.aspx';
$data = array(
    'username' => 'your_username',
    'password' => 'your_password',
    'msgid' => '12345',
    'dstaddr' => '0912345678'
);

$options = array(
    'http' => array(
        'header' => "Content-type: application/x-www-form-urlencoded\r\n",
        'method' => 'POST',
        'content' => http_build_query($data)
    )
);

$context = stream_context_create($options);
$result = file_get_contents($url, false, $context);

if ($result === FALSE) {

```

```

    /* Handle error */
    echo "Error sending request.";
}

var_dump($result);

?>

```

Python

```

import requests

url = 'https://www.smsgo.com.tw/sms_gw/query.aspx'
payload = {
    'username': 'your_username',
    'password': 'your_password',
    'msgid': '12345',
    'dstaddr': '0912345678'
}

try:
    response = requests.post(url, data=payload)
    response.raise_for_status() # Raise an exception for bad status codes
    print("Response from server:")
    print(response.text)
except requests.exceptions.RequestException as e:
    print(f"An error occurred: {e}")

```

Java

```

import java.net.URI;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;

public class SmsGoExample {
    public static void main(String[] args) throws Exception {
        HttpClient client = HttpClient.newHttpClient();
        HttpRequest request = HttpRequest.newBuilder()
            .uri(URI.create("https://www.smsgo.com.tw/sms_gw/query.aspx"))
            .header("Content-Type", "application/x-www-form-urlencoded")
            .POST(HttpRequest.BodyPublishers.ofString("username=your_username&password=your_password&msgid=12345&dstaddr=0912345678"))
            .build();

        HttpResponse<String> response = client.send(request, HttpResponse.BodyHandlers.ofString());
        System.out.println(response.body());
    }
}

```

GET

/sms_gw/queryBulk.aspx

Query Bulk SMS Status (GET) ■ Deprecated

■ **DEPRECATED**

■ **Deprecated** — Use POST instead.

Query delivery status of multiple SMS. msgid and dstaddr are comma-separated and counts must match.

Parameters

Name	In	Type	Required	Description
username	query	string	Yes	Member account username
password	query	string	Yes	Member password or API Key
msgid	query	string	Yes	One or more message IDs separated by commas. Count must match dstaddr.
dstaddr	query	string	Yes	One or more phone numbers separated by commas. Count must match msgid.
rtype	query	string (enum)	No	Response format. Omit for plain text, JSON for JSON, XML for XML.

Responses

200

Multiple records separated by \r\n. Each record: statusCode|statusStr|cellphone|msgid

text/plain — Success

```
0|DELIVRD|0912345678|100101121234567
4|REJECTD|0987654321|100101121234568
```

text/plain — Serial number not found

```
-2|-2|can not find this serial number!
```

application/json — Success

```
{
  "results": {
    "result": [
      {
        "statusCode": "0",
        "statusStr": "DELIVRD",
        "cellphone": "0912345678",
        "msgid": "100101121234567"
      },
      {
        "statusCode": "4",
        "statusStr": "REJECTD",
        "cellphone": "0987654321",
        "msgid": "100101121234568"
      }
    ]
  }
}
```

application/json — Serial number not found

```
{
  "result": {
    "msgid": "-1",
    "statusCode": "-1",
    "statusstr": "CGI string error!!",
    "point": 0
  }
}
```

application/xml — Success

```
<results><result><statusCode>0</statusCode><statusStr>DELIVRD</statusStr><cellphone>0912345678</cellphone><msgid>100101121234567</msgid></result><result><statusCode>4</statusCode><statusStr>REJECTD</statusStr><cellphone>0987654321</cellphone><msgid>100101121234568</msgid></result></results>
```

application/xml — Serial number not found

```
<results><r><statusCode>-2</statusCode><statusStr>can not find this serial number!</statusStr></r></results>
```

Code Samples

cURL

```
curl "https://www.smsgo.com.tw/sms_gw/queryBulk.aspx?username=your_username&password=your_password&msgid=12345,12346&dstaddr=0912345678,0987654321"
```

JavaScript

```
fetch('https://www.smsgo.com.tw/sms_gw/queryBulk.aspx?username=your_username&password=your_password&msgid=12345,12346&dstaddr=0912345678,0987654321', {
  method: 'GET'
})
.then(response => response.text())
.then(result => console.log(result))
.catch(error => console.log('error', error));
```

C#

```
using System;
using System.Net.Http;
using System.Threading.Tasks;

class Program
{
    static async Task Main(string[] args)
    {
        var client = new HttpClient();
        var request = new HttpRequestMessage(HttpMethod.Get, "https://www.smsgo.com.tw/sms_gw/queryBulk.aspx?username=your_username&password=your_password&msgid=12345,12346&dstaddr=0912345678,0987654321");
        var response = await client.SendAsync(request);
        response.EnsureSuccessStatusCode();
        Console.WriteLine(await response.Content.ReadAsStringAsync());
    }
}
```

PHP

```
<?php

$url = 'https://www.smsgo.com.tw/sms_gw/queryBulk.aspx?username=your_username&password=your_password&msgid=12345,12346&dstaddr=0912345678,0987654321';
$result = file_get_contents($url);

if ($result === FALSE) {
    echo "Error sending request.";
}

var_dump($result);

?>
```

Python

```
import requests

url = 'https://www.smsgo.com.tw/sms_gw/queryBulk.aspx?username=your_username&password=your_password&msgid=12345,12346&dstaddr=0912345678,0987654321'

try:
    response = requests.get(url)
    response.raise_for_status()
    print(response.text)
except requests.exceptions.RequestException as e:
    print(f"An error occurred: {e}")
```

Java

```
import java.net.URI;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;

public class SmsGoExample {
    public static void main(String[] args) throws Exception {
        HttpClient client = HttpClient.newHttpClient();
        HttpRequest request = HttpRequest.newBuilder()
            .uri(URI.create("https://www.smsgo.com.tw/sms_gw/queryBulk.aspx?username=your_username&password=your_password&msgid=12345,12346&dstaddr=0912345678,0987654321"))
            .GET()
            .build();
        HttpResponse<String> response = client.send(request, HttpResponse.BodyHandlers.ofString());
        System.out.println(response.body());
    }
}
```

POST

/sms_gw/queryBulk.aspx

Query Bulk SMS Status (POST)

Query delivery status of multiple SMS via POST. msgid and dstaddr are comma-separated and counts must match. Accepts application/x-www-form-urlencoded or application/json.

Request Body

application/x-www-form-urlencoded

application/json

Responses

200

Multiple records separated by \r\n. Each record: statusCode|statusStr|cellphone|msgid

text/plain — Success

```
0|DELIVRD|0912345678|100101121234567
4|REJECTD|0987654321|100101121234568
```

text/plain — Serial number not found

```
-2|-2|can not find this serial number!
```

application/json — Success

```
{
  "results": {
    "result": [
      {
        "statusCode": "0",
        "statusStr": "DELIVRD",
        "cellphone": "0912345678",
        "msgid": "100101121234567"
      }
    ]
  }
}
```

application/json — Serial number not found

```
{
  "result": {
    "msgid": "-1",
    "statusCode": "-1",
    "statusstr": "CGI string error!!",
    "point": 0
  }
}
```

application/xml — Success

```
<results><result><statusCode>0</statusCode><statusStr>DELIVRD</statusStr><cellphone>0912345678</cellphone><msgid>100101121234567</msgid></result><result><statusCode>4</statusCode><statusStr>REJECTD</statusStr><cellphone>0987654321</cellphone><msgid>100101121234568</msgid></result></results>
```

application/xml — Serial number not found

```
<results><r><statusCode>-2</statusCode><statusStr>can not find this serial number!</statusStr></r></results>
```

Code Samples

cURL

```
curl -X POST -d 'username=your_username&password=your_password&msgid=12345,12346&dstaddr=0912345678,0987654321' https://www.smsgo.com.tw/sms_gw/queryBulk.aspx
```

JavaScript

```
const myHeaders = new Headers();
myHeaders.append("Content-Type", "application/x-www-form-urlencoded");

const urlencoded = new URLSearchParams();
urlencoded.append("username", "your_username");
urlencoded.append("password", "your_password");
urlencoded.append("msgid", "12345,12346");
urlencoded.append("dstaddr", "0912345678,0987654321");

const requestOptions = {
  method: 'POST',
  headers: myHeaders,
  body: urlencoded,
  redirect: 'follow'
};

fetch("https://www.smsgo.com.tw/sms_gw/queryBulk.aspx", requestOptions)
  .then(response => response.text())
```

```
.then(result => console.log(result))
.catch(error => console.log('error', error));
```

C#

```
using System;
using System.Collections.Generic;
using System.Net.Http;
using System.Threading.Tasks;

public class SmsGoExample
{
    public static async Task Main(string[] args)
    {
        var client = new HttpClient();
        var values = new Dictionary<string, string>
        {
            { "username", "your_username" },
            { "password", "your_password" },
            { "msgid", "12345,12346" },
            { "dstaddr", "0912345678,0987654321" }
        };

        var content = new FormUrlEncodedContent(values);
        var response = await client.PostAsync("https://www.smsgo.com.tw/sms_gw/queryBulk.aspx"
, content);
        var responseString = await response.Content.ReadAsStringAsync();

        Console.WriteLine(responseString);
    }
}
```

PHP

```
<?php

$url = 'https://www.smsgo.com.tw/sms_gw/queryBulk.aspx';
$data = array(
    'username' => 'your_username',
    'password' => 'your_password',
    'msgid' => '12345,12346',
    'dstaddr' => '0912345678,0987654321'
);

$options = array(
    'http' => array(
        'header' => "Content-type: application/x-www-form-urlencoded\r\n",
        'method' => 'POST',
        'content' => http_build_query($data)
    )
);

$context = stream_context_create($options);
$result = file_get_contents($url, false, $context);

if ($result === FALSE) {
    echo "Error sending request.";
}

var_dump($result);

?>
```

Python

```
import requests

url = 'https://www.smsgo.com.tw/sms_gw/queryBulk.aspx'
payload = {
    'username': 'your_username',
    'password': 'your_password',
    'msgid': '12345,12346',
    'dstaddr': '0912345678,0987654321'
}

try:
    response = requests.post(url, data=payload)
    response.raise_for_status()
    print(response.text)
except requests.exceptions.RequestException as e:
    print(f"An error occurred: {e}")
```

Java

```
import java.net.URI;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;

public class SmsGoExample {
    public static void main(String[] args) throws Exception {
        HttpClient client = HttpClient.newHttpClient();
        HttpRequest request = HttpRequest.newBuilder()
            .uri(URI.create("https://www.smsgo.com.tw/sms_gw/queryBulk.aspx"))
            .header("Content-Type", "application/x-www-form-urlencoded")
            .POST(HttpRequest.BodyPublishers.ofString("username=your_username&password=your_password&msgid=12345,12346&dstaddr=0912345678,0987654321"))
            .build();
        HttpResponse<String> response = client.send(request, HttpResponse.BodyHandlers.ofString());
        System.out.println(response.body());
    }
}
```

POST

/sms_gw/queryReply.aspx

Query SMS Replies (POST)

Retrieve inbound reply messages sent by recipients within a date range. Accepts application/x-www-form-urlencoded or application/json.

Request Body

application/x-www-form-urlencoded

application/json

Responses

200

List of reply messages.

text/plain — Success

```
0912345678|0800123456|2026/06/01 10:05:00|2026/06/01 09:00:00|■■■■■
```

text/plain — Error

```
-2|-2|Error! Username, password or IP is wrong. Please retry again!
```

application/json

```
{
  "results": {
    "result": [
      {
        "srcaddr": "0912345678",
        "dstaddr": "0800123456",
        "recvtime": "2026/06/01 10:05:00",
        "sendtime": "2026/06/01 09:00:00",
        "smbody": "■■■■■"
      }
    ]
  }
}
```

application/xml

```
<results><result><srcaddr>0912345678</srcaddr><dstaddr>0800123456</dstaddr><recvtime>2026/06/01 10:05:00</recvtime><sendtime>2026/06/01 09:00:00</sendtime><smbody>■■■■■</smbody></result></results>
```

Code Samples

cURL

```
curl -X POST -d 'username=your_username&password=your_password&startdate=2026/06/01&enddate=2026/06/30&rtype=JSON' https://www.smsgo.com.tw/sms_gw/queryReply.aspx
```

JavaScript

```
const params = new URLSearchParams();
params.append('username', 'your_username');
params.append('password', 'your_password');
params.append('startdate', '2026/06/01');
params.append('enddate', '2026/06/30');
params.append('rtype', 'JSON');

fetch('https://www.smsgo.com.tw/sms_gw/queryReply.aspx', {
  method: 'POST',
  body: params
})
.then(r => r.json())
.then(console.log)
.catch(console.error);
```

C#

```
using System;
using System.Collections.Generic;
using System.Net.Http;
using System.Threading.Tasks;

public class SmsGoExample
{
    public static async Task Main(string[] args)
    {
        var client = new HttpClient();
```

```

var values = new Dictionary<string, string>
{
    { "username", "your_username" },
    { "password", "your_password" },
    { "startdate", "2026/06/01" },
    { "enddate", "2026/06/30" },
    { "rtype", "JSON" }
};
var content = new FormUrlEncodedContent(values);
var response = await client.PostAsync("https://www.smsgo.com.tw/sms_gw/queryReply.aspx", content);
Console.WriteLine(await response.Content.ReadAsStringAsync());
}
}

```

PHP

```

<?php
$url = 'https://www.smsgo.com.tw/sms_gw/queryReply.aspx';
$data = ['username'=>'your_username', 'password'=>'your_password', 'startdate'=>'2026/06/01', 'enddate'=>'2026/06/30', 'rtype'=>'JSON'];
$options = ['http'=>['header'=>"Content-type: application/x-www-form-urlencoded\r\n", 'method'=>'POST', 'content'=>http_build_query($data)]];
$result = file_get_contents($url, false, stream_context_create($options));
var_dump($result);
?>

```

Python

```

import requests

url = 'https://www.smsgo.com.tw/sms_gw/queryReply.aspx'
payload = {
    'username': 'your_username',
    'password': 'your_password',
    'startdate': '2026/06/01',
    'enddate': '2026/06/30',
    'rtype': 'JSON'
}
response = requests.post(url, data=payload)
print(response.json())

```

Java

```

import java.net.URI;
import java.net.http.*;

public class SmsGoExample {
    public static void main(String[] args) throws Exception {
        HttpClient client = HttpClient.newHttpClient();
        HttpRequest request = HttpRequest.newBuilder()
            .uri(URI.create("https://www.smsgo.com.tw/sms_gw/queryReply.aspx"))
            .header("Content-Type", "application/x-www-form-urlencoded")
            .POST(HttpRequest.BodyPublishers.ofString("username=your_username&password=your_password&startdate=2026/06/01&enddate=2026/06/30&rtype=JSON"))
            .build();
        HttpResponse<String> response = client.send(request, HttpResponse.BodyHandlers.ofString());
        System.out.println(response.body());
    }
}

```

Query Sent Message List (POST)

Retrieve a paginated list of sent messages within a date range (max 31 days). Accepts application/x-www-form-urlencoded or application/json.

Request Body

application/x-www-form-urlencoded

application/json

Responses

200

Paginated list of sent messages.

text/plain — Success

```
page=1|pagesize=50|totalcount=1|totalpages=1
S2026060109000001|0912345678|2026/06/01 09:00:00|Hello!|0|DELIVRD
```

text/plain — Error

```
-2|-2|Error! Username, password or IP is wrong. Please retry again!
```

application/json

```
{
  "results": {
    "page": 1,
    "pagesize": 50,
    "totalcount": 1,
    "totalpages": 1,
    "result": [
      {
        "msgid": "S2026060109000001",
        "dstaddr": "0912345678",
        "sendtime": "2026/06/01 09:00:00",
        "smbdy": "Hello from SMSGO!",
        "statuscode": "0",
        "statusstr": "DELIVRD"
      }
    ]
  }
}
```

application/xml

```
<results><page>1</page><pagesize>50</pagesize><totalcount>1</totalcount><totalpages>1</totalpages><result><msgid>S2026060109000001</msgid><dstaddr>0912345678</dstaddr><sendtime>2026/06/01 09:00:00</sendtime><smbdy>Hello from SMSGO!</smbdy><statuscode>0</statuscode><statusstr>DELIVRD</statusstr></result></results>
```

Code Samples

cURL

```
curl -X POST -d 'username=your_username&password=your_password&startdate=2026/06/01&enddate=2026/06/30&page=1&pagesize=50&rtype=JSON' https://www.smsgo.com.tw/sms_gw/querySentList.aspx
```

JavaScript

```

const params = new URLSearchParams();
params.append('username', 'your_username');
params.append('password', 'your_password');
params.append('startdate', '2026/06/01');
params.append('enddate', '2026/06/30');
params.append('page', '1');
params.append('pagesize', '50');
params.append('rtype', 'JSON');

fetch('https://www.smsgo.com.tw/sms_gw/querySentList.aspx', {
  method: 'POST',
  body: params
})
.then(r => r.json())
.then(console.log)
.catch(console.error);

```

C#

```

using System;
using System.Collections.Generic;
using System.Net.Http;
using System.Threading.Tasks;

public class SmsGoExample
{
    public static async Task Main(string[] args)
    {
        var client = new HttpClient();
        var values = new Dictionary<string, string>
        {
            { "username", "your_username" },
            { "password", "your_password" },
            { "startdate", "2026/06/01" },
            { "enddate", "2026/06/30" },
            { "page", "1" },
            { "pagesize", "50" },
            { "rtype", "JSON" }
        };
        var content = new FormUrlEncodedContent(values);
        var response = await client.PostAsync("https://www.smsgo.com.tw/sms_gw/querySentList.aspx", content);
        Console.WriteLine(await response.Content.ReadAsStringAsync());
    }
}

```

PHP

```

<?php
$url = 'https://www.smsgo.com.tw/sms_gw/querySentList.aspx';
$data = ['username'=>'your_username', 'password'=>'your_password', 'startdate'=>'2026/06/01', 'enddate'=>'2026/06/30', 'page'=>1, 'pagesize'=>50, 'rtype'=>'JSON'];
$options = ['http'=>['header'=>"Content-type: application/x-www-form-urlencoded\r\n", 'method'=>'POST', 'content'=>http_build_query($data)]];
$result = file_get_contents($url, false, stream_context_create($options));
var_dump($result);
?>

```

Python

```

import requests

url = 'https://www.smsgo.com.tw/sms_gw/querySentList.aspx'
payload = {

```

```

        'username': 'your_username',
        'password': 'your_password',
        'startdate': '2026/06/01',
        'enddate': '2026/06/30',
        'page': 1,
        'pagesize': 50,
        'rtype': 'JSON'
    }
    response = requests.post(url, data=payload)
    print(response.json())

```

Java

```

import java.net.URI;
import java.net.http.*;

public class SmsGoExample {
    public static void main(String[] args) throws Exception {
        HttpClient client = HttpClient.newHttpClient();
        HttpRequest request = HttpRequest.newBuilder()
            .uri(URI.create("https://www.smsgo.com.tw/sms_gw/querySentList.aspx"))
            .header("Content-Type", "application/x-www-form-urlencoded")
            .POST(HttpRequest.BodyPublishers.ofString("username=your_username&password=your_password&startdate=2026/06/01&enddate=2026/06/30&page=1&pagesize=50&rtype=JSON"))
            .build();
        HttpResponse<String> response = client.send(request, HttpResponse.BodyHandlers.ofString());
        System.out.println(response.body());
    }
}

```

Account

Operations for managing account settings and points.

GET /sms_gw/queryPoint.aspx

Query Remaining Points (GET) ■ Deprecated

■ **DEPRECATED**

■ **Deprecated** — Use POST instead.

Retrieve remaining SMS point balance.

Parameters

Name	In	Type	Required	Description
username	query	string	Yes	Member account username
password	query	string	Yes	Member password or API Key
rtype	query	string (enum)	No	Response format. Omit for plain text, JSON for JSON, XML for XML.

Responses

200

Remaining points. Plain text returns a decimal number (e.g. 6220.0).

text/plain — Success

```
6220.0
```

text/plain — Error

```
Error! User name, password, serial number or ip is wrong. Please retry again!
```

application/json

```
{
  "results": {
    "result": [
      {
        "statusCode": "0",
        "point": "6220.0"
      }
    ]
  }
}
```

application/xml

```
<results><result><statusCode>0</statusCode><point>6220.0</point></result></results>
```

Code Samples

cURL

```
curl "https://www.smsgo.com.tw/sms_gw/queryPoint.aspx?username=your_username&password=your_password"
```

JavaScript

```

fetch('https://www.smsgo.com.tw/sms_gw/queryPoint.aspx?username=your_username&password=your_password', {
  method: 'GET'
})
.then(response => response.text())
.then(result => console.log(result))
.catch(error => console.log('error', error));

```

C#

```

using System;
using System.Net.Http;
using System.Threading.Tasks;

class Program
{
    static async Task Main(string[] args)
    {
        var client = new HttpClient();
        var request = new HttpRequestMessage(HttpMethod.Get, "https://www.smsgo.com.tw/sms_gw/queryPoint.aspx?username=your_username&password=your_password");
        var response = await client.SendAsync(request);
        response.EnsureSuccessStatusCode();
        Console.WriteLine(await response.Content.ReadAsStringAsync());
    }
}

```

PHP

```

<?php

$url = 'https://www.smsgo.com.tw/sms_gw/queryPoint.aspx?username=your_username&password=your_password';
$result = file_get_contents($url);

if ($result === FALSE) {
    echo "Error sending request.";
}

var_dump($result);

?>

```

Python

```

import requests

url = 'https://www.smsgo.com.tw/sms_gw/queryPoint.aspx?username=your_username&password=your_password'

try:
    response = requests.get(url)
    response.raise_for_status()
    print(response.text)
except requests.exceptions.RequestException as e:
    print(f"An error occurred: {e}")

```

Java

```

import java.net.URI;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;

```

```
public class SmsGoExample {
    public static void main(String[] args) throws Exception {
        HttpClient client = HttpClient.newHttpClient();
        HttpRequest request = HttpRequest.newBuilder()
            .uri(URI.create("https://www.smsgo.com.tw/sms_gw/queryPoint.aspx?username=your_username&password=your_password"))
            .GET()
            .build();
        HttpResponse<String> response = client.send(request, HttpResponse.BodyHandlers.ofString());
        System.out.println(response.body());
    }
}
```

POST

/sms_gw/queryPoint.aspx

Query Remaining Points (POST)

Retrieve remaining SMS point balance via POST. Accepts application/x-www-form-urlencoded or application/json.

Request Body

application/x-www-form-urlencoded

application/json

Responses

200

Remaining points. Plain text: decimal number (e.g. 6220.0).

text/plain — Success

6220.0

text/plain — Error

Error! User name, password, serial number or ip is wrong.

application/json

```
{
  "results": {
    "result": [
      {
        "statusCode": "0",
        "point": "6220.0"
      }
    ]
  }
}
```

application/xml

```
<results><result><statusCode>0</statusCode><point>6220.0</point></result></results>
```

Code Samples

cURL

```
curl -X POST -d 'username=your_username&password=your_password' https://www.smsgo.com.tw/sms_gw/queryPoint.aspx
```

JavaScript

```
const myHeaders = new Headers();
myHeaders.append("Content-Type", "application/x-www-form-urlencoded");

const urlencoded = new URLSearchParams();
urlencoded.append("username", "your_username");
urlencoded.append("password", "your_password");

const requestOptions = {
  method: 'POST',
  headers: myHeaders,
  body: urlencoded,
  redirect: 'follow'
};

fetch("https://www.smsgo.com.tw/sms_gw/queryPoint.aspx", requestOptions)
  .then(response => response.text())
  .then(result => console.log(result))
  .catch(error => console.log('error', error));
```

C#

```
using System;
using System.Collections.Generic;
using System.Net.Http;
using System.Threading.Tasks;

public class SmsGoExample
{
    public static async Task Main(string[] args)
    {
        var client = new HttpClient();
        var values = new Dictionary<string, string>
        {
            { "username", "your_username" },
            { "password", "your_password" }
        };

        var content = new FormUrlEncodedContent(values);
        var response = await client.PostAsync("https://www.smsgo.com.tw/sms_gw/queryPoint.aspx", content);
        var responseString = await response.Content.ReadAsStringAsync();

        Console.WriteLine(responseString);
    }
}
```

PHP

```
<?php

$url = 'https://www.smsgo.com.tw/sms_gw/queryPoint.aspx';
$data = array(
    'username' => 'your_username',
    'password' => 'your_password'
);

$options = array(
    'http' => array(
        'header' => "Content-type: application/x-www-form-urlencoded\r\n",
```

```

        'method' => 'POST',
        'content' => http_build_query($data)
    )
);
$context = stream_context_create($options);
$result = file_get_contents($url, false, $context);

if ($result === FALSE) {
    echo "Error sending request.";
}

var_dump($result);

?>

```

Python

```

import requests

url = 'https://www.smsgo.com.tw/sms_gw/queryPoint.aspx'
payload = {
    'username': 'your_username',
    'password': 'your_password'
}

try:
    response = requests.post(url, data=payload)
    response.raise_for_status()
    print(response.text)
except requests.exceptions.RequestException as e:
    print(f"An error occurred: {e}")

```

Java

```

import java.net.URI;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;

public class SmsGoExample {
    public static void main(String[] args) throws Exception {
        HttpClient client = HttpClient.newHttpClient();
        HttpRequest request = HttpRequest.newBuilder()
            .uri(URI.create("https://www.smsgo.com.tw/sms_gw/queryPoint.aspx"))
            .header("Content-Type", "application/x-www-form-urlencoded")
            .POST(HttpRequest.BodyPublishers.ofString("username=your_username&password=your_password"))
            .build();
        HttpResponse<String> response = client.send(request, HttpResponse.BodyHandlers.ofString());
        System.out.println(response.body());
    }
}

```

POST

/sms_gw/moWebHook.aspx

Set / Get / Clear MO Webhook

Register, retrieve, or remove the global MO (Mobile Originated) Webhook URL for your account.

When an end-user replies to an SMS, SMSGO calls this URL via HTTP GET with the following query parameters:

Parameter	Description
srcaddr	Sender's mobile number
dstaddr	Destination (virtual) number
smbody	Reply message content (URL-encoded)
timestamp	Received time (yyMMddHHmmss)
msgid	Message identifier

This is a global fallback webhook — it fires when no per-send `replyurl` was set for the matched send.

Request Body

`application/x-www-form-urlencoded`

`application/json`

Responses

200

MO Webhook result. `statuscode=0` means success.

Error Codes:

Code	Description
0	Success
-2	Username, password, or IP error
-3	Username or password empty
-11	Database error
-1	Invalid URL format

text/plain — Success

```
0
https://abc.trycloudflare.com/mo_receive
```

text/plain — Error

```
-3
```

application/json — Success

```
{
  "results": {
    "result": [
      {
        "statuscode": "0",
        "webhookurl": "https://abc.trycloudflare.com/mo_receive"
      }
    ]
  }
}
```

application/json — Error

```
{
  "results": {
```

```

"result": [
{
"statusCode": "-3",
"webhookurl": ""
}
]
}
}

```

application/xml — Success

```

<results><result><statusCode>0</statusCode><webhookurl>https://abc.trycloudflare.com/mo_receive</webhookurl></result></results>

```

application/xml — Error

```

<results><result><statusCode>-3</statusCode><webhookurl></webhookurl></result></results>

```

Code Samples

cURL

```

curl -X POST -d 'username=your_username&password=your_password&action=set&webhookurl=https://abc.trycloudflare.com/mo_receive' https://www.smsgo.com.tw/sms_gw/moWebHook.aspx

```

JavaScript

```

const urlencoded = new URLSearchParams();
urlencoded.append('username', 'your_username');
urlencoded.append('password', 'your_password');
urlencoded.append('action', 'set');
urlencoded.append('webhookurl', 'https://abc.trycloudflare.com/mo_receive');

fetch('https://www.smsgo.com.tw/sms_gw/moWebHook.aspx', {
  method: 'POST',
  headers: { 'Content-Type': 'application/x-www-form-urlencoded' },
  body: urlencoded
})
.then(r => r.text()).then(console.log);

```

C#

```

var client = new HttpClient();
var values = new Dictionary<string, string> {
    { "username", "your_username" },
    { "password", "your_password" },
    { "action", "set" },
    { "webhookurl", "https://abc.trycloudflare.com/mo_receive" }
};
var response = await client.PostAsync("https://www.smsgo.com.tw/sms_gw/moWebHook.aspx", new FormUrlEncodedContent(values));
Console.WriteLine(await response.Content.ReadAsStringAsync());

```

PHP

```

<?php
$url = 'https://www.smsgo.com.tw/sms_gw/moWebHook.aspx';
$data = ['username'=>'your_username', 'password'=>'your_password', 'action'=>'set', 'webhookurl'=>'https://abc.trycloudflare.com/mo_receive'];
$content = stream_context_create(['http'=>['method'=>'POST', 'header'=>'Content-type: application/x-www-form-urlencoded', 'content'=>http_build_query($data)]];
echo file_get_contents($url, false, $context);
?>

```

Python

```
import requests
response = requests.post('https://www.smsgo.com.tw/sms_gw/moWebHook.aspx', data={
    'username': 'your_username',
    'password': 'your_password',
    'action': 'set',
    'webhookurl': 'https://abc.trycloudflare.com/mo_receive'
})
print(response.text)
```

Java

```
import java.net.URI;
import java.net.http.*;
public class Example {
    public static void main(String[] args) throws Exception {
        HttpClient client = HttpClient.newHttpClient();
        HttpRequest request = HttpRequest.newBuilder()
            .uri(URI.create("https://www.smsgo.com.tw/sms_gw/moWebHook.aspx"))
            .header("Content-Type", "application/x-www-form-urlencoded")
            .POST(HttpRequest.BodyPublishers.ofString("username=your_username&password=your_password&action=set&webhookurl=https://abc.trycloudflare.com/mo_receive"))
            .build();
        System.out.println(client.send(request, HttpResponse.BodyHandlers.ofString()).body());
    }
}
```

Verification

Operations for sending and verifying one-time passwords (OTP).

GET `/sms_gw/verify.aspx`

Send OTP Code (GET) ■ Deprecated

■ **DEPRECATED**

■ **Deprecated** — Use POST instead.

Send a one-time password (OTP) to a mobile number.

OTP Flow Step 1/2: Call this → save msgid from response → proceed to `/verifyAck.aspx`.

Parameters

Name	In	Type	Required	Description
username	query	string	Yes	Member account username
password	query	string	Yes	Member password or API Key
dstaddr	query	string	Yes	Recipient mobile number(s). Multiple numbers comma-separated (max 50). Domestic: 09xxxxxxx. International: prefix with + URL-encoded as %2b, e.g. %2b8613681912700.
codeLength	query	integer (enum)	No	Length of the OTP verification code. Can be 4 or 6. Default: 4.
domain	query	string	No	Optional custom domain for iOS/Android OTP autofill (e.g., shopee.tw). Must be pre-approved.
dlvtime	query	string	No	Scheduled delivery time. Format: YYYY/MM/DD hh24:mm:ss. Leave blank to send immediately.
response	query	string	No	Callback URL for delivery status reports. System POSTs status updates to this URL. Leave blank to disable.
rtype	query	string (enum)	No	Response format. Omit for plain text, JSON for JSON, XML for XML.

Responses

200

API response. Format determined by `rtype` parameter. `statusCode=0` means success.

Error Codes:

Code	Description
0	Success
-1	CGI string error
-2	Username, password, or IP error
-3	Username or password empty
-5	Phone number format error
-8	Insufficient points

-9	Too many recipients (max 50)
-10	Send failed
-11	Database error
-12	Message body empty
-13	Parameter count mismatch
-14	Fraud keyword detected
-15	IP address not allowed
-16	API service not enabled
-17	Callback URL not allowed
-18	Off-net number
-19	Number blocked
-20	Marked as SPAM
-21	Capacity reached
-22	Request rerouted
-23	No signature
-24	OTP verification failed
-25	OTP expired
-30	Operator system ID error

text/plain — OTP sent successfully

```
msgid=2601270523943266
statuscode=0
statusstr=Verify success
point=1
```

text/plain — Authentication error

```
msgid=-3
statuscode=-3
statusstr=username or password empty!
point=0
```

application/json — OTP sent successfully

```
{
  "result": {
    "msgid": "2601270523943266",
    "statuscode": "0",
    "statusstr": "Verify success",
    "point": 1
  }
}
```

application/json — Authentication error

```
{
  "result": {
    "msgid": "-3",
    "statuscode": "-3",
    "statusstr": "username or password empty!",
    "point": 0
  }
}
```

application/xml — OTP sent successfully

```
<response><code>0</code><msgid>2601270523943266</msgid><text>Verify success</text></response>
```

application/xml — Authentication error

```
<response><code>-3</code><msgid>-3</msgid><text>username or password empty!</text></response>
```

Code Samples

cURL

```
curl "https://www.smsgo.com.tw/sms_gw/verify.aspx?username=your_username&password=your_password&dstaddr=0912345678"
```

JavaScript

```
fetch('https://www.smsgo.com.tw/sms_gw/verify.aspx?username=your_username&password=your_password&dstaddr=0912345678', {
  method: 'GET'
})
.then(response => response.text())
.then(result => console.log(result))
.catch(error => console.log('error', error));
```

C#

```
using System;
using System.Net.Http;
using System.Threading.Tasks;

class Program
{
    static async Task Main(string[] args)
    {
        var client = new HttpClient();
        var request = new HttpRequestMessage(HttpMethod.Get, "https://www.smsgo.com.tw/sms_gw/verify.aspx?username=your_username&password=your_password&dstaddr=0912345678");
        var response = await client.SendAsync(request);
        response.EnsureSuccessStatusCode();
        Console.WriteLine(await response.Content.ReadAsStringAsync());
    }
}
```

PHP

```
<?php

$url = 'https://www.smsgo.com.tw/sms_gw/verify.aspx?username=your_username&password=your_password&dstaddr=0912345678';
$result = file_get_contents($url);

if ($result === FALSE) {
    echo "Error sending request.";
}

var_dump($result);

?>
```

Python

```
import requests

url = 'https://www.smsgo.com.tw/sms_gw/verify.aspx?username=your_username&password=your_password&dstaddr=0912345678'

try:
```

```

        response = requests.get(url)
        response.raise_for_status()
        print(response.text)
except requests.exceptions.RequestException as e:
    print(f"An error occurred: {e}")

```

Java

```

import java.net.URI;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;

public class SmsGoExample {
    public static void main(String[] args) throws Exception {
        HttpClient client = HttpClient.newHttpClient();
        HttpRequest request = HttpRequest.newBuilder()
            .uri(URI.create("https://www.smsgo.com.tw/sms_gw/verify.aspx?username=your_username&password=your_password&dstaddr=0912345678"))
            .GET()
            .build();
        HttpResponse<String> response = client.send(request, HttpResponse.BodyHandlers.ofString());
        System.out.println(response.body());
    }
}

```

POST

/sms_gw/verify.aspx

Send OTP Code (POST)

Send a one-time password (OTP) to a mobile number. Accepts application/x-www-form-urlencoded or application/json.

OTP Flow Step 1/2: Call this → save msgid from response → user enters code → call /verifyAck.aspx with serial_number=msgid.

Request Body

application/x-www-form-urlencoded

application/json

Responses

200

API response. Format determined by rtype parameter. statuscode=0 means success.

Error Codes:

Code	Description
0	Success
-1	CGI string error
-2	Username, password, or IP error
-3	Username or password empty

-5	Phone number format error
-8	Insufficient points
-9	Too many recipients (max 50)
-10	Send failed
-11	Database error
-12	Message body empty
-13	Parameter count mismatch
-14	Fraud keyword detected
-15	IP address not allowed
-16	API service not enabled
-17	Callback URL not allowed
-18	Off-net number
-19	Number blocked
-20	Marked as SPAM
-21	Capacity reached
-22	Request rerouted
-23	No signature
-24	OTP verification failed
-25	OTP expired
-30	Operator system ID error

text/plain — OTP sent successfully

```
msgid=2601270523943266
statuscode=0
statusstr=Verify success
point=1
```

text/plain — Authentication error

```
msgid=-3
statuscode=-3
statusstr=username or password empty!
point=0
```

application/json — OTP sent successfully

```
{
  "result": {
    "msgid": "2601270523943266",
    "statuscode": "0",
    "statusstr": "Verify success",
    "point": 1
  }
}
```

application/json — Authentication error

```
{
  "result": {
    "msgid": "-3",
    "statuscode": "-3",
    "statusstr": "username or password empty!",
    "point": 0
  }
}
```

```
}
```

application/xml — OTP sent successfully

```
<response><code>0</code><msgid>2601270523943266</msgid><text>Verify success</text></response>
```

application/xml — Authentication error

```
<response><code>-3</code><msgid>-3</msgid><text>username or password empty!</text></response>
```

Code Samples

cURL

```
curl -X POST -d 'username=your_username&password=your_password&dstaddr=0912345678' https://www.smsgo.com.tw/sms_gw/verify.aspx
```

JavaScript

```
const myHeaders = new Headers();
myHeaders.append("Content-Type", "application/x-www-form-urlencoded");

const urlencoded = new URLSearchParams();
urlencoded.append("username", "your_username");
urlencoded.append("password", "your_password");
urlencoded.append("dstaddr", "0912345678");

const requestOptions = {
  method: 'POST',
  headers: myHeaders,
  body: urlencoded,
  redirect: 'follow'
};

fetch("https://www.smsgo.com.tw/sms_gw/verify.aspx", requestOptions)
  .then(response => response.text())
  .then(result => console.log(result))
  .catch(error => console.log('error', error));
```

C#

```
using System;
using System.Collections.Generic;
using System.Net.Http;
using System.Threading.Tasks;

public class SmsGoExample
{
    public static async Task Main(string[] args)
    {
        var client = new HttpClient();
        var values = new Dictionary<string, string>
        {
            { "username", "your_username" },
            { "password", "your_password" },
            { "dstaddr", "0912345678" }
        };

        var content = new FormUrlEncodedContent(values);
        var response = await client.PostAsync("https://www.smsgo.com.tw/sms_gw/verify.aspx", content);
        var responseString = await response.Content.ReadAsStringAsync();

        Console.WriteLine(responseString);
    }
}
```

```
}  
}
```

PHP

```
<?php  
  
$url = 'https://www.smsgo.com.tw/sms_gw/verify.aspx';  
$data = array(  
    'username' => 'your_username',  
    'password' => 'your_password',  
    'dstaddr' => '0912345678'  
);  
  
$options = array(  
    'http' => array(  
        'header' => "Content-type: application/x-www-form-urlencoded\r\n",  
        'method' => 'POST',  
        'content' => http_build_query($data)  
    )  
);  
$context = stream_context_create($options);  
$result = file_get_contents($url, false, $context);  
  
if ($result === FALSE) {  
    echo "Error sending request."  
}  
  
var_dump($result);  
  
?>
```

Python

```
import requests  
  
url = 'https://www.smsgo.com.tw/sms_gw/verify.aspx'  
payload = {  
    'username': 'your_username',  
    'password': 'your_password',  
    'dstaddr': '0912345678'  
}  
  
try:  
    response = requests.post(url, data=payload)  
    response.raise_for_status()  
    print(response.text)  
except requests.exceptions.RequestException as e:  
    print(f"An error occurred: {e}")
```

Java

```
import java.net.URI;  
import java.net.http.HttpClient;  
import java.net.http.HttpRequest;  
import java.net.http.HttpResponse;  
  
public class SmsGoExample {  
    public static void main(String[] args) throws Exception {  
        HttpClient client = HttpClient.newHttpClient();  
        HttpRequest request = HttpRequest.newBuilder()  
            .uri(URI.create("https://www.smsgo.com.tw/sms_gw/verify.aspx"))  
            .header("Content-Type", "application/x-www-form-urlencoded")
```

```

        .POST(HttpRequest.BodyPublishers.ofString("username=your_username&password=your_password&dstaddr=0912345678"))
        .build();
    HttpResponse<String> response = client.send(request, HttpResponse.BodyHandlers.ofString());
    System.out.println(response.body());
}
}

```

GET /sms_gw/verifyAck.aspx

Verify OTP Code (GET) ■ Deprecated

■ DEPRECATED

■ **Deprecated** — Use POST instead.

OTP Flow Step 2/2: Validate the code entered by the user.

Parameters

Name	In	Type	Required	Description
username	query	string	Yes	Member account username
password	query	string	Yes	Member password or API Key
dstaddr	query	string	Yes	Recipient mobile number(s). Multiple numbers comma-separated (max 50). Domestic: 09xxxxxxx. International: prefix with + URL-encoded as %2b, e.g. %2b8613681912700.
OtpCode	query	string	Yes	The OTP code entered by the user (4 or 6 digits).
serial_number	query	string	Yes	The msgid returned from the Send OTP (/verify.aspx) request.
resend	query	boolean	No	Set to true to automatically resend a new OTP if the current one has expired (statuscode -25).
dlvtime	query	string	No	Scheduled delivery time. Format: YYYY/MM/DD hh24:mm:ss. Leave blank to send immediately.
response	query	string	No	Callback URL for delivery status reports. System POSTs status updates to this URL. Leave blank to disable.
rtype	query	string (enum)	No	Response format. Omit for plain text, JSON for JSON, XML for XML.

Responses

200

API response. Format determined by rtype parameter. statuscode=0 means success.

Error Codes:

Code	Description
0	Success
-1	CGI string error

-2	Username, password, or IP error
-3	Username or password empty
-5	Phone number format error
-8	Insufficient points
-9	Too many recipients (max 50)
-10	Send failed
-11	Database error
-12	Message body empty
-13	Parameter count mismatch
-14	Fraud keyword detected
-15	IP address not allowed
-16	API service not enabled
-17	Callback URL not allowed
-18	Off-net number
-19	Number blocked
-20	Marked as SPAM
-21	Capacity reached
-22	Request rerouted
-23	No signature
-24	OTP verification failed
-25	OTP expired
-30	Operator system ID error

text/plain — OTP verified successfully

```
msgid=2603160424021870
statuscode=0
statusstr=OK
point=0
```

text/plain — Wrong OTP code (-24)

```
msgid=-24
statuscode=-24
statusstr=Verify Fail
point=0
```

text/plain — OTP expired (-25)

```
msgid=-25
statuscode=-25
statusstr=Verify Expired
point=0
```

application/json — OTP verified successfully

```
{
  "result": {
    "msgid": "2603160424021870",
    "statuscode": "0",
    "statusstr": "OK",
    "point": 0
  }
}
```

application/json — Wrong OTP code (-24)

```
{
  "result": {
    "msgid": "-24",
    "statusCode": "-24",
    "statusstr": "Verify Fail",
    "point": 0
  }
}
```

application/json — OTP expired (-25)

```
{
  "result": {
    "msgid": "-25",
    "statusCode": "-25",
    "statusstr": "Verify Expired",
    "point": 0
  }
}
```

application/xml — OTP verified successfully

```
<response><code>0</code><msgid>2603160424021870</msgid><text>OK</text></response>
```

application/xml — Wrong OTP code (-24)

```
<response><code>-24</code><msgid>-24</msgid><text>Verify Fail</text></response>
```

application/xml — OTP expired (-25)

```
<response><code>-25</code><msgid>-25</msgid><text>Verify Expired</text></response>
```

Code Samples

cURL

```
curl "https://www.smsgo.com.tw/sms_gw/verifyAck.aspx?username=your_username&password=your_password&dstaddr=0912345678&otpCode=123456&serial_number=1234567890"
```

JavaScript

```
fetch('https://www.smsgo.com.tw/sms_gw/verifyAck.aspx?username=your_username&password=your_password&dstaddr=0912345678&otpCode=123456&serial_number=1234567890', {
  method: 'GET'
})
.then(response => response.text())
.then(result => console.log(result))
.catch(error => console.log('error', error));
```

C#

```
using System;
using System.Net.Http;
using System.Threading.Tasks;

class Program
{
    static async Task Main(string[] args)
    {
        var client = new HttpClient();
        var request = new HttpRequestMessage(HttpMethod.Get, "https://www.smsgo.com.tw/sms_gw/verifyAck.aspx?username=your_username&password=your_password&dstaddr=0912345678&otpCode=123456&serial_number=1234567890");
        var response = await client.SendAsync(request);
        response.EnsureSuccessStatusCode();
    }
}
```

```

        Console.WriteLine(await response.Content.ReadAsStringAsync());
    }
}

```

PHP

```

<?php

$url = 'https://www.smsgo.com.tw/sms_gw/verifyAck.aspx?username=your_username&password=your_password&dstaddr=0912345678&otpCode=123456&serial_number=1234567890';
$result = file_get_contents($url);

if ($result === FALSE) {
    echo "Error sending request.";
}

var_dump($result);

?>

```

Python

```

import requests

url = 'https://www.smsgo.com.tw/sms_gw/verifyAck.aspx?username=your_username&password=your_password&dstaddr=0912345678&otpCode=123456&serial_number=1234567890'

try:
    response = requests.get(url)
    response.raise_for_status()
    print(response.text)
except requests.exceptions.RequestException as e:
    print(f"An error occurred: {e}")

```

Java

```

import java.net.URI;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;

public class SmsGoExample {
    public static void main(String[] args) throws Exception {
        HttpClient client = HttpClient.newHttpClient();
        HttpRequest request = HttpRequest.newBuilder()
            .uri(URI.create("https://www.smsgo.com.tw/sms_gw/verifyAck.aspx?username=your_username&password=your_password&dstaddr=0912345678&otpCode=123456&serial_number=1234567890"))
            .GET()
            .build();
        HttpResponse<String> response = client.send(request, HttpResponse.BodyHandlers.ofString());
        System.out.println(response.body());
    }
}

```

POST

/sms_gw/verifyAck.aspx

Verify OTP Code (POST)

OTP Flow Step 2/2: Validate the OTP code entered by the user. Accepts application/x-www-form-urlencoded or application/json.

- serial_number = msgid from the Send OTP response
- otpCode = 4 or 6 digit code the user received
- statuscode -24 = wrong code, -25 = expired (set resend=true to auto-resend)

Request Body

application/x-www-form-urlencoded

application/json

Responses

200

API response. Format determined by rtype parameter. statuscode=0 means success.

Error Codes:

Code	Description
0	Success
-1	CGI string error
-2	Username, password, or IP error
-3	Username or password empty
-5	Phone number format error
-8	Insufficient points
-9	Too many recipients (max 50)
-10	Send failed
-11	Database error
-12	Message body empty
-13	Parameter count mismatch
-14	Fraud keyword detected
-15	IP address not allowed
-16	API service not enabled
-17	Callback URL not allowed
-18	Off-net number
-19	Number blocked
-20	Marked as SPAM
-21	Capacity reached
-22	Request rerouted
-23	No signature
-24	OTP verification failed
-25	OTP expired
-30	Operator system ID error

text/plain — OTP verified successfully

```
msgid=2603160424021870
statuscode=0
statusstr=OK
```

```
point=0
```

text/plain — Wrong OTP code (-24)

```
msgid=-24
statuscode=-24
statusstr=Verify Fail
point=0
```

text/plain — OTP expired (-25)

```
msgid=-25
statuscode=-25
statusstr=Verify Expired
point=0
```

application/json — OTP verified successfully

```
{
  "result": {
    "msgid": "2603160424021870",
    "statuscode": "0",
    "statusstr": "OK",
    "point": 0
  }
}
```

application/json — Wrong OTP code (-24)

```
{
  "result": {
    "msgid": "-24",
    "statuscode": "-24",
    "statusstr": "Verify Fail",
    "point": 0
  }
}
```

application/json — OTP expired (-25)

```
{
  "result": {
    "msgid": "-25",
    "statuscode": "-25",
    "statusstr": "Verify Expired",
    "point": 0
  }
}
```

application/xml — OTP verified successfully

```
<response><code>0</code><msgid>2603160424021870</msgid><text>OK</text></response>
```

application/xml — Wrong OTP code (-24)

```
<response><code>-24</code><msgid>-24</msgid><text>Verify Fail</text></response>
```

application/xml — OTP expired (-25)

```
<response><code>-25</code><msgid>-25</msgid><text>Verify Expired</text></response>
```

Code Samples

cURL

```
curl -X POST -d 'username=your_username&password=your_password&dstaddr=0912345678&otpCode=123456&serial_number=1234567890' https://www.smsgo.com.tw/sms_gw/verifyAck.aspx
```

JavaScript

```
const myHeaders = new Headers();
myHeaders.append("Content-Type", "application/x-www-form-urlencoded");

const urlencoded = new URLSearchParams();
urlencoded.append("username", "your_username");
urlencoded.append("password", "your_password");
urlencoded.append("dstaddr", "0912345678");
urlencoded.append("OtpCode", "123456");
urlencoded.append("serial_number", "1234567890");

const requestOptions = {
  method: 'POST',
  headers: myHeaders,
  body: urlencoded,
  redirect: 'follow'
};

fetch("https://www.smsgo.com.tw/sms_gw/verifyAck.aspx", requestOptions)
  .then(response => response.text())
  .then(result => console.log(result))
  .catch(error => console.log('error', error));
```

C#

```
using System;
using System.Collections.Generic;
using System.Net.Http;
using System.Threading.Tasks;

public class SmsGoExample
{
    public static async Task Main(string[] args)
    {
        var client = new HttpClient();
        var values = new Dictionary<string, string>
        {
            { "username", "your_username" },
            { "password", "your_password" },
            { "dstaddr", "0912345678" },
            { "OtpCode", "123456" },
            { "serial_number", "1234567890" }
        };

        var content = new FormUrlEncodedContent(values);
        var response = await client.PostAsync("https://www.smsgo.com.tw/sms_gw/verifyAck.aspx", content);
        var responseString = await response.Content.ReadAsStringAsync();

        Console.WriteLine(responseString);
    }
}
```

PHP

```
<?php

$url = 'https://www.smsgo.com.tw/sms_gw/verifyAck.aspx';
$data = array(
    'username' => 'your_username',
    'password' => 'your_password',
    'dstaddr' => '0912345678',
```

```

        'OtpCode' => '123456',
        'serial_number' => '1234567890'
    );

    $options = array(
        'http' => array(
            'header' => "Content-type: application/x-www-form-urlencoded\r\n",
            'method' => 'POST',
            'content' => http_build_query($data)
        )
    );
    $context = stream_context_create($options);
    $result = file_get_contents($url, false, $context);

    if ($result === FALSE) {
        echo "Error sending request.";
    }

    var_dump($result);

?>

```

Python

```

import requests

url = 'https://www.smsgo.com.tw/sms_gw/verifyAck.aspx'
payload = {
    'username': 'your_username',
    'password': 'your_password',
    'dstaddr': '0912345678',
    'OtpCode': '123456',
    'serial_number': '1234567890'
}

try:
    response = requests.post(url, data=payload)
    response.raise_for_status()
    print(response.text)
except requests.exceptions.RequestException as e:
    print(f"An error occurred: {e}")

```

Java

```

import java.net.URI;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;

public class SmsGoExample {
    public static void main(String[] args) throws Exception {
        HttpClient client = HttpClient.newHttpClient();
        HttpRequest request = HttpRequest.newBuilder()
            .uri(URI.create("https://www.smsgo.com.tw/sms_gw/verifyAck.aspx"))
            .header("Content-Type", "application/x-www-form-urlencoded")
            .POST(HttpRequest.BodyPublishers.ofString("username=your_username&password=your_password&dstaddr=0912345678&OtpCode=123456&serial_number=1234567890"))
            .build();
        HttpResponse<String> response = client.send(request, HttpResponse.BodyHandlers.ofString());
        System.out.println(response.body());
    }
}

```

```
}
```
