

## Overview

The Voxox SMS HTTP API gives you the ability to:

- Send SMS messages
- Check on delivery status
- Delivery Receipts (DLR) posted over web callback HTTP POST
- Inbound SMS delivered over web callback

## Getting Started

Voxox requires an IP address or network (Eg. 216.151.151.0/24) to authorize for the use of the API. On request, we can remove this requirement but it is not recommended.

If the API is authorized for use from any IP address, the customer must acknowledge that they understand and are accountable for the risk associated.

To get started, the below 3 required items will be needed.

Parameter	Required	Description
username	Yes	<b>Voxox</b> to provide SMPP/API account login
password	Yes	<b>Voxox</b> to provide SMPP/API account password
IP address	Yes	<b>Partner</b> to provide the IP address or network to authorize to make API requests
DLR callback URL	No	<b>Partner</b> to provide URL to POST Delivery Receipts
Inbound SMS callback URL	No	<b>Partner</b> to provide URL to POST inbound SMS messages

## Outbound SMS

Sending SMS with the Voxox HTTP API supports the same standards as our SMPP direct connections. We will cover the following two examples in detail below

### Basic

- Single part Messages less than 160 characters
- GSM encoding

### Advanced

- Multipart messages over 160 characters
- Unicode/UCS-2 or GSM encoding
- [KB article](#) for a detailed description of GSM-7 vs. UCS-2 Encoding

## SMS Submit - Basic

The following is the most simple method to send an SMS that is under 160 characters and using default encoding.

### Example URL

HTTP PUT or GET method

```
https://admin.sms.voxox.ws:8002/api?username=<username>&password=<password>&ani=18885551234&dnis=16195551212&command=submit&message=your%20TP%20code&longMessageMode=split
```

Parameter	Required	Type	Description
username	Yes	string	Your SMPP/API account login
password	Yes	string	Your SMPP/API account password
ani	Yes	integer	From Number
dnis	Yes	integer	To Number
command	Yes	string	Set this to 'submit' when sending SMS
message	Yes	string	URL Encoded Message Body
longMessageMode	No	string	Set this to 'split' to send long messages

**Note:** If you do not set longMessageMode=split, then messages greater than 160 characters will be cut and only the first 160 characters will be delivered.

### **Example Return**

message\_id, json encoded.

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=UTF-8

{"message_id":"5ddf8aac-7eb8-f8f9-8ba4-bfd62659ed5e"}
```

**Note** see next section for example response for multipart messages

## SMS Submit - Advanced

The example below builds upon the basic one above. Added to this example is UCS-2/Unicode encoding and multipart messages which results from the message being greater than 67 characters long.

For more information on message encoding and lengths, please refer to our article [SMS Encoding: GSM-7 & Unicode/UCS-2](#)

### Example URL

HTTP PUT or GET method

```
https://admin.sms.voxox.ws:8002/api?username=<username>&password=<password>&ani=18885551234&dnis=16195551212&message=your%20OTP%20code%20is%3A%2012345%20%20&command=submit&longMessageMode=split&dataCoding=8
```

Parameter	Required	Type	Description
username	Yes	string	Your SMPP/API account login
password	Yes	string	Your SMPP/API account password
ani	Yes	integer	From Number
dnis	Yes	integer	To Number
command	Yes	string	Set this to 'submit' when sending SMS
message	Yes	string	URL Encoded Message Body
longMessageMode	No	string	Set this to 'split' to send long messages
dataCoding	No	integer	0, 8 (GSM, UCS-2)
incMsgId	No	string	Your internal message Id

### Example Return

message\_id, json encoded.

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=UTF-8

[{"dnis":"16195551212","message_id":"5ddf8aac-44d3-4b6d-ba70-c512bb42607d","segment_num":"1"}, {"dnis":"16195551212","message_id":"5ddf8aac-57ab-dc7b-4909-bc853a686077","segment_num":"2"}]
```

### PHP + cURL Example

- Replace username, password, ani and dnis with your values

```
<?php
// Set Variables
$smsUrl      = "https://admin.sms.voxox.ws:8002/api?";
$login      = "<username>";
$pass       = "<password>";
$ani        = "18885551234";
$dnis       = "16195551212";
$body       = "your OTP code is: 12345";
$body       = urlencode($body); // URL Encode Body
$datacoding = "8";
$mode       = "split";

// All HTTP POST Data
$data="ani={$ani}&dnis={$dnis}&body={$body}&username={$login}&password={$pass}&data
Coding={$datacoding}&longMessageMode={$mode}&command=submit&message={$body}";

// CURL.
$ch = curl_init();

curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $data);
curl_setopt($ch, CURLOPT_URL, $smsUrl);

$result = curl_exec ($ch) or die(curl_error($ch));
curl_close ($ch);

echo $result;
?>
```

## Messages Status

There are two ways to check on the status of the message you sent. The first is using the 'query' command. Like the 'submit' example above, the query command is a HTTP PUT or GET method. Please find an example below.

### Query command

#### Example URL

HTTP PUT or GET method

```
https://admin.sms.voxox.ws:8002/api?username=<username>&password=<password>&message_id=<message_id>&command=query
```

Parameter	Required	Type	Description
username	Yes	string	Your SMPP/API account login
password	Yes	string	Your SMPP/API account password
message_id	Yes	string	Message ID returned from submit command
command	Yes	string	Set this to 'query' when looking up a record

#### Example Return

Status, delivery\_time, mccmnc; json encoded.

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=UTF-8

{"status": "DELIVRD", "delivery_time": "1912191706", "mccmnc": "310012"}
```

#### Possible Delivery STATUS responses

DELIVRD: Message is delivered to the Subscriber  
ENROUTE: Message is in routing stage  
SENT: Message is delivered to our perring partner, pending delivery receipt  
UNDELIV: Message cannot be delivered  
UNKNOWN: Unknown message status  
REJECTD: Message was rejected by SMSC

## DLR Callback URL

Voxox also offers a callback URL option if you would like to receive automatic Delivery Receipt (DLR) receipt. You can configure your service with a callback URL to have DLR updates pushed to you, instead of having to query our API for a message status update.

The parameters below are the default values and may be configured upon request. For example, if you would like 'ani' to be 'from' please contact support to change the value.

### Array Key Description

- All types are string

Parameter	Description
message_id	message identifier, including the common ID of a concatenated Message for multipart messages
ani	From number (url encoded)
dnis	To number
message	Delivery report text (url encoded)
delivery_status	Delivery status (see 'Possible Delivery STATUS responses' above)
result_code	Code from the delivery report, passed from our peering partner (default value is '000')
delivery_time	Time when the delivery report was generated (YYMMDDHHMMSS in GMT+0)

## Inbound SMS

### Overview

If you are using 2 way SMS, then you will need to provide a callback URL where you will listen for a HTTP POST.

All incoming messages destined to any of your provisioned numbers will be relayed to a predefined Callback URL.

### Inbound Callback URL

A POST to the Callback URL gets triggered once a message destined to one of your DIDs is received by our SMS Center.

The POST request will have a single parameter, called data, data is a JSON encoded array. The array will contain the message envelope information and its contents.

### Inbound Array Key Description

- All types are string

Parameter	Description
message_id	message identifier, including the common ID of a concatenated message for multipart messages
ani	From number (URL encoded)
dnis	To number
message	Delivery report text (URL encoded)
delivery_status	Delivery status (see 'Possible Delivery STATUS responses' above)
result_code	Code from the delivery report, passed from our peering partner (default value is '000')
delivery_time	Time when the delivery report was generated (YYMMDDHHMMSS in GMT+0)

**Note:** Depending on the service provider of your number, you may receive a %2B (URL encoded +) prepending the ANI. Safe to assume that the ani should be decoded.