



**GTX Mobile Messaging SMS Gateway Interface  
General SMS REST API Manual**

**SMS API v. 2.0**

Revision: 03  
Revision Date: 6/5/19 4:01:00 PM  
Authors: Oliver Zabel

## Introduction

The Mobile Messaging SMS Gateway provides an easy-to-use and easy-to-implement REST API for submitting SMS to and receiving DLR from the gateway.

This manual will describe the interface and provide some examples on how to use this interface. If you have further questions regarding this interface, please contact your account manager or file a trouble ticket to support@gtx-messaging.com.

## Sending SMS

### Endpoint

You can reach the SMS Gateway under the following URL using HTTPs on port 443:

```
https://rest.gtx-messaging.net/smsc/sendsms/:auth-key[/:format]
```

**Note:** Due to data protection reasons the API is only accessible via HTTPs with TLS versions 1.0, 1.1 or 1.2.

### Path Parameters

Path Parameter	Mandatory	Type	Description	Example
<b>auth-key</b>	yes	String	Your personal API Key, provided in GTX Dashboard	„aaaaaaaa-bbbb-cccc-dddd-1234567890ab“
<b>format</b>	no	String	Sets the output format for the SMSC response. Allowed: “json”, “xml”, “plain”. Default is “plain”	“json”

### HTTP Methods

The allowed request methods for sending SMS are GET, POST or PUSH.

### HTTP Headers (optional)

In order to set the format of the parameters transferred to the General SMS REST API or the wished output format, please choose the corresponding HTTP headers “Accept” and “Content-Type”.

Key	Value	Mandatory	Type	Description	Example
<b>Accept</b>	application/x-www-form-urlencoded	no	String	Set “plain” output format	“Accept: application/x-www-form-urlencoded”
<b>Accept</b>	application/json	no	String	Set “json” output format	“Accept: application/json”
<b>Content-Type</b>	application/x-www-form-urlencoded	no	String	Sets plain format for the request parameters	“Content-Type: application/x-www-form-urlencoded”
<b>Content-Type</b>	application/json	no	String	Sets json format for the request parameters	“Content-Type: application/json”

**Note:** Default Content-Type as input format is for “Content-Type: application/x-www-form-urlencoded”.

## Request Parameters

The following HTTP-GET, HTTP-POST or HTTP-PUSH parameters can be set to submit a message:

Field	Mandatory	Type	Description	Example
<b>from</b>	yes	String	The TPOA / originator of the message. Allowed is alphanumeric up to 11 chars, shortcode, local longcode or international number (E.164, E.212 or E.214)	"Company" "+49171000000" "01729000000" "55888"
<b>to</b>	yes	String	The recipient of the message, international format, with leading "+" (E.164, E.212 or E.214)	"+49171000000"
<b>text</b>	yes	String	Content of the message	"Hello World"
<b>dlr-mask</b>	no	Number	Request for delivery reports with the state of the sent message. (bit mask) The value is a bit mask composed of: 1: Delivered to phone 2: Non-Delivered to Phone 34: Expired 66: Unknown	"3" for Status "DELIVRD", UNDELIV" and "EXPIRED"
<b>dlr-url</b>	no	String	If dlr-mask is given, this mandatory URL will be fetched by HTTP-GET method.	"https://mydomain.com:12345/dlr?"
<b>udh</b>	no	String	User Data Header (UDH) part of the message, HEX encoded.	"0605040B8423F0"
<b>dcs</b>	no	Number	Data Coding Scheme (DCS) in HEX	"0" for text msg, "8" for binary
<b>mclass</b>	no	Number	Message Class; accepted values: 0 to 3	"0" for submit display
<b>mwi</b>	no	Number	Message Waiting Indicator (MWI) sets the MWI bits in DCS	
<b>coding</b>	no	Number	Sets the coding scheme bits in DCS field. Accepts values 1 to 3	"1" 7-Bit Message, "2" 8-Bit Message, "3" UCS-2
<b>charset</b>	no	String	Encoding of the message parameter	"UTF-16"
<b>validity</b>	no	Number	Validity Period (VP) in minutes	"240"
<b>validity-time</b>	no	ISO-8601	Validity Date Time	"2019-05-01T12:00:00Z"
<b>deferred</b>	no	Number	Deferred Delivery Time (DDT) in minutes	"5"
<b>deferred-time</b>	no	ISO-8601	Deferred Delivery Date Time	"2019-05-01T12:00:00Z"

## Response:

As a result you will receive following HTTP status codes and body:

HTTP Status	HTTP Code	Response Fields	Retriable
<b>HTTP_OK</b>	200	<message-count> <message-status> <message-id>	NO
<b>HTTP_BAD_REQUEST</b>	400	Error list for details	NO
<b>HTTP_UNAUTHORIZED</b>	401	Error list for details	NO
<b>HTTP_FORBIDDEN</b>	403	Error list for details	NO
<b>HTTP_INTERNAL_SERVER_ERROR</b>	500	Error list for details	NO

**Please note:** GTX General SMS REST API gateway accepts long messages and split them where needed. Hence please always read **<message-count>** returned by gateway.

## Delivery Notification (DLR):

Description of the parameters signalling delivery reports (DLR):

Field	Type	Description
<b>from</b>	String	Phone number of the receiver.
<b>to</b>	String	Phone number of the sender.
<b>message-id</b>	String	Message ID of the submitted message
<b>dlr-mask</b>	Number	State of the sent message. (bit mask) The value is a bit mask composed of: 1: Delivered to phone 2: Non-Delivered to Phone 34: Expired 66: Unknown
<b>error-code</b>	Number	Error code
<b>error-message</b>	String	Error message
<b>submit-date</b>	ISO-8601	Timestamp of submission
<b>done-date</b>	ISO-8601	Timestamp of delivery
<b>plmn</b>	Number	Public Land Mobile Number (PLMN): Networkcode (MCC+MNC) of the sent SMS
<b>country</b>	ISO 3166-1 alpha-2	two-letter country code of the destination of the SMS
<b>sms-cost</b>	Decimal	SMS price charged

### Notes:

- 1) **“from”** and **“to”** fields are swapped in delivery reports
- 2) Client always has to respond with **HTTP\_OK** and body: **„OK“**

## Examples:

### 1) Successful Sending:

URL for sending one message and request delivery reports to signal on 'https://www.mydomain.com:12345/dlr\_backchannel?'

HTTP-Get-Request with response:

```
https://rest.gtx-messaging.net/smsc/sendsms/aaaaaaaa-bbbb-cccc-dddd-1234567890ab/json?from=test&to=%2b4917xxxxxxx&text=test&dlr-mask=3&dlr-url=https%3A%2F%2Fwww.mydomain.com%3A12345%2Fdlr_backchannel%3F
```

Or UCS2 Request:

```
https://rest.gtx-messaging.net/smsc/sendsms/aaaaaaaa-bbbb-cccc-dddd-1234567890ab/json?from=test&to=%2b4917xxxxxxx&text=просто+тест&coding=3&charset=utf-8&dlr_mask=3&dlr_url=https%3A%2F%2Fwww.mydomain.com%3A12345%2Fdlr_backchannel%3F
```

The same UCS2 Request without charset parameter (text in UCS2):

```
https://rest.gtx-messaging.net/smsc/sendsms/aaaaaaaa-bbbb-cccc-dddd-1234567890ab/json?from=test&to=%2b491729084747&text=%04%3f%04%40%04%3e%04%41%04%42%04%3e%00%20%04%42%04%35%04%41%04%42&coding=3&dlr-mask=7&dlr-url=https%3A%2F%2Fwww.mydomain.com%3A12345%2Fdlr_backchannel%3F
```

GTX gateway response *HTTP\_OK*, the result is JSON formatted

```
{"message-count": "1", "message-status": "OK", "message-id": "44ca4150-7382-4e17-91a6-599628dca12e"}
```

Client Response *HTTP\_OK*:

```
OK
```

### 2. Failed Sending

Request ("from"-parameter missing):

```
https://rest.gtx-messaging.net/smsc/sendsms/aaaaaaaa-bbbb-cccc-dddd-1234567890ab/json?to=%2b491729084747&text=test&dlr-mask=3&dlr-url=https%3A%2F%2Fwww.mydomain.com%3A12345%2Fdlr_backchannel%3F
```

GTX gateway response *HTTP\_BAD\_REQUEST*:

```
{"from": ["can't be blank", "is too short (minimum is 1 character)"]}
```