

SMPP API Overview

Voxox SMS over SMPP gives you the ability to send and receive SMS messages using the SMPP protocol.

→ Support for **SMPP Version 3.4** only.

Connection Information

SMPP Server

The SMPP Server resides at: **smpp.sms.voxox.com**

SMPP Port

To connect using SMPP you need to connect to the server above and **use the port documented in your welcome letter**. This is usually a number in the 14000-14999 range.

System ID/Type

In order to successfully bind to our SMPP servers, you need to specify the system-type SMPP parameter, **this value is documented in your welcome letter**. It usually consists of the word SMPP and the SMPP Port number. E.g. SMPP14000

Bind Mode

Our binds are set up as transceivers, so you'll only need to bind once and set the bind to transceiver mode.

DLR Visibility

Long Code

Since we act as an SMSC aggregator, we don't have visibility into the full message life-cycle, instead we provide DLR visibility to our gateway. Specifically, when setting the 'registered_delivery' flag of the 'submit_sm' SMPP PDU to 1, we will respond with bitmask 8 for smsc submit or bitmask 16 for smsc reject to acknowledge the receipt of the message by our gateway and either bitmask 1 for delivery success or bitmask 2 for delivery failure when the message has reached to our peering partners, not the handset.

Short Code

For shortcode services we provide full visibility. Specifically, when setting the 'registered_delivery' flag of the 'submit_sm' SMPP PDU to 1, we will respond with either bitmask 1 for delivery success or bitmask 2 for delivery failure.

Supported Message Codings and Charsets

Supported Characters with data_coding = 0

!#\$%&'()*+,-./0123456789;<=>?@ABCDEFGHIJKLMNQRSTUvwxyz[]^_`abcdefghijklmnopqrstuvwxyz{|}~

Example SMPP PDU:

```
DEBUG: SMPP PDU 0x7f47f8000a40 dump:
DEBUG: type_name: submit_sm
DEBUG: command_id: 4 = 0x00000004
DEBUG: command_status: 0 = 0x00000000
DEBUG: sequence_number: 86 = 0x00000056
DEBUG: service_type: NULL
```

```
DEBUG: source_addr_ton: 2 = 0x00000002
DEBUG: source_addr_npi: 1 = 0x00000001
DEBUG: source_addr: "16198195520"
DEBUG: dest_addr_ton: 2 = 0x00000002
DEBUG: dest_addr_npi: 1 = 0x00000001
DEBUG: destination_addr: "18587366685"
DEBUG: esm_class: 3 = 0x00000003
DEBUG: protocol_id: 0 = 0x00000000
DEBUG: priority_flag: 0 = 0x00000000
DEBUG: schedule_delivery_time: NULL
DEBUG: validity_period: NULL
DEBUG: registered_delivery: 0 = 0x00000000
DEBUG: replace_if_present_flag: 0 = 0x00000000
DEBUG: data_coding: 0 = 0x00000000
DEBUG: sm_default_msg_id: 0 = 0x00000000
DEBUG: sm_length: 102 = 0x00000066
DEBUG: short_message:
DEBUG: Octet string at 0x7f47f8001780:
DEBUG: len: 102
DEBUG: size: 1024
DEBUG: immutable: 0
DEBUG: data: 21 22 23 02 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 !"#$%&'()+,.-./0
DEBUG: data: 31 32 33 34 35 36 37 38 39 3a 3b 3c 3d 3e 3f 00 123456789;<=>?.
DEBUG: data: 41 42 43 44 45 46 47 48 49 4a 4b 4c 4d 4e 4f 50 ABCDEFGHIJKLMNOP
DEBUG: data: 51 52 53 54 55 56 57 58 59 5a 1b 3c 1b 2f 1b 3e QRSTUVWXYZ.<./.>
DEBUG: data: 1b 14 11 3f 61 62 63 64 65 66 67 68 69 6a 6b 6c ...?abcdefghijklmnopkl
DEBUG: data: 6d 6e 6f 70 71 72 73 74 75 76 77 78 79 7a 1b 28 mnopqrstuvwxyz.(
DEBUG: data: 1b 40 1b 29 1b 3d .@.) .=
DEBUG: Octet string dump ends.
DEBUG: SMPP PDU dump ends.
```

UTF-8 is supported with data_coding = 8

Example SMPP PDU: (string: Γελά)

```
DEBUG: SMPP PDU 0x7f47f8000a40 dump:
DEBUG: type_name: submit_sm
DEBUG: command_id: 4 = 0x00000004
DEBUG: command_status: 0 = 0x00000000
DEBUG: sequence_number: 106 = 0x0000006a
DEBUG: service_type: NULL
DEBUG: source_addr_ton: 2 = 0x00000002
DEBUG: source_addr_npi: 1 = 0x00000001
DEBUG: source_addr: "16198195520"
DEBUG: dest_addr_ton: 2 = 0x00000002
DEBUG: dest_addr_npi: 1 = 0x00000001
DEBUG: destination_addr: "18587366685"
DEBUG: esm_class: 3 = 0x00000003
DEBUG: protocol_id: 0 = 0x00000000
DEBUG: priority_flag: 0 = 0x00000000
DEBUG: schedule_delivery_time: NULL
DEBUG: validity_period: NULL
DEBUG: registered_delivery: 0 = 0x00000000
DEBUG: replace_if_present_flag: 0 = 0x00000000
DEBUG: data_coding: 8 = 0x00000008
DEBUG: sm_default_msg_id: 0 = 0x00000000
DEBUG: sm_length: 21 = 0x00000015
DEBUG: short_message:
DEBUG: Octet string at 0x7f47f8001e90:
DEBUG: len: 21
DEBUG: size: 22
DEBUG: immutable: 0
DEBUG: data: d0 b4 d0 be d0 b1 d1 80 d0 be d0 b5 20 d1 83 d1 .....
```

```
DEBUG: data: 82 d1 80 d0 be .....  
DEBUG: Octet string dump ends.  
DEBUG: SMPP PDU dump ends.
```

Supported Message Classes.

Some of our routes support message class bits of DCS: 0 (directly to display, flash). All of our routes support message class bits of DCS: 1 (to mobile).

Message Concatenation.

Most of our routes, support message concatenation of up to 10 messages.

Supported SMPP PDUs

- bind_transmitter
- bind_receiver
- bind_transceiver
- unbind
- enquire_link
- data_sm
- submit_sm
- deliver_sm_resp
- unbind_resp

Examples

Kannel

We provide Kannel configuration examples. Kannel is one of the most popular Open Source SMSC software. For commercial SMPP clients, please contact your vendor or refer to their documentation.

Configuration

```
group=smc  
smc=smpp  
smc-id=voxoxTXRX  
interface-version=34  
host=smpp.sms.voxox.com  
port=your-smpp-port  
smc-username=your-smpp-username  
smc-password=your-smpp-password  
system-type=your-smpp-system-id  
transceiver-mode=1
```