SMSC WebService User Guide

Modified:	2024-11-05
Version:	3.4

1 Index

2 Change history

Rev	Date	By	Changes from previous release
1.0	2009-04-02	KCN	Created
1.1	2009-12-16	KCN	Added customParameters to SendParameters Added lookup method Added response code 200 (OK)
2.0	2013-04-05	KCN	Updated support for long SMS Added error codes Changed logo
2.0a	2013-06-25	KCN	Minor updates of spelling and added more custom parameters.
3.0	2014-09-18	JA	New method: Refund Added result codes for refunds
3.0a	2020-08-28	KCN	TLS 1.2 Appendix
3.1	2021-05-10	EP	URLs to WSDL Locations changed and refund references removed
3.2	2024-11-05	FS	Added result codes
3.3	2025-03-06	GM	Added result codes and URL:s
3.4	2025-07-03	KCN	Updated Appendix 1, updated TLS information and supported ciphers.

3 Introduction

LINK Mobility has been a SMS distributor since 2001 and has much experience in working with both operators and connection aggregators. This platform is designed to handle large traffic volumes, maintain a high availability and make it easy to route traffic via multiple connections.

This is document describes the WebService (SOAP) interface to the SMSC-platform.

This document will not handle specific use cases as concatenated messages, WAP-push, Flash SMS, etc. More information about those cases can be provided by contacting support.

4 WSDL and locations

This interface can be reached at two different URLs which are connected to the Internet via different ISPs. You can choose to access this service via HTTP or HTTPS.

We recommend that you implement your program so it automatically can switch between both URLs to get higher availability.

Protocol	**URL to WSDL**
HTTPS 1	https://n-eu.linkmobility.io/smscws/api?wsdl
HTTPS 2	https://n-eu.linkmobility.io/smscws/api?wsdl
HTTPS 1	https://c-eu.linkmobility.io/smscws/api?wsdl
HTTPS 2	https://c-eu.linkmobility.io/smscws/api?wsdl
HTTPS 1	https://s-eu.linkmobility.io/smscws/api?wsdl
HTTPS 2	https://s-eu.linkmobility.io/smscws/api?wsdl

5 Methods in the WebService

The methods are listed below with their parameters and all possible response codes.

5.1 getVersion

This method returns the current version of the WebService.

5.1.1 Parameters

There are no parameters for this method.

5.1.2 Returns

This method returns the current version number of the WebService as a string. This value should only be used for information or testing of availability of the service.

5.2 send

This is the method that should be used when sending SMS messages.

5.2.1 Parameters

Parameter	Data type	Description
message	SendParameters	See SendParameters for more information.

5.2.2 Returns

This method returns an object of the type SendResult.

5.3 lookup

This method lookup which operator a specific number will be delivered to.

5.3.1 Parameters

Parameter	Data type	Description
message	LookupParameters	See LookupParameters for more information.

5.3.2 Returns

This method returns an object of the type LookupResult. Successful results with supplied operator will use return code 200 (OK).

5.4 Objects in the WebService

The following objects are used in the WebService.

5.5 SendParameters

Name	Data type	Description
username	String	This is the username which is used for authentication. This is provided by Support.
serviceId	Integer	This is the serviceId which is used for authentication. This is provided by Support.
password	String	This is the password which is used for authentication. This is provided by Support.
source	String	This is the source number from where the message should be sent. The format is depending on the specified sourceTON.
sourceTON	Integer	This is the source type of number. See TON for more information.
destination	String	This is the destination number. The format is depending on the specified destinationTON.
destinationTON	Integer	This is the destination type of number. See TON for more information.
dcs	Integer	This is the Data Coding Scheme that should be used when sending the SMS. See DCS for more information.
userDataHeader	String	This value may be specified when sending concatenated SMS, WAP-push, etc. The format is hex encoded 8-bit bytes. More information about valid UDH for long SMS etc may be given by support upon request.
userData	String	This is the message itself. The DCS specifies the format on this value. - GSM default alphabet encoded messages has a maximum length of 1377 bytes. Note that Extended GSM characters need 2 bytes for one character. 1 GSM7 message is 160 non-extended characters. 153 non-extended characters for GSM7 if the message is concatenated. - Binary messages should be hex encoded as 8-bit bytes and the maximum length is 140 bytes (280 bytes when hex encoded). - UCS2 encoded messages has a maximum length of 567

		characters. - Note that messages will be split into several SMS if the text doesn't fit in one SMS. 1 UCS2 encoded message is 70 characters or 63 characters if the message is concatenated.
useDeliveryReport	Boolean	True indicates that a delivery report should be sent back when the message has come to a final state. It's recommended to set this value to true.
validityTime	Long	This specifies how long the message is supposed to live. The value should be specified in milliseconds. -1 indicates default validity time should apply. Recommended time is between 15 minutes (900000) and 48 hours (172800000).
tariffClass	String	Used for Premium SMS otherwise empty or null. Format: <currency in ISO 4217><cent of currency> Example: SEK350 equals 3.50 SEK
vat	Float	Used for Premium SMS. 25 equals 25% -1 equals not used or default.
customerParameters	List of CustomParameter	Optional, parameters may be specified if requested by support. List of available constants.

5.6 SendResult

Name	Data type	Description
messageId	String	This is the unique messageId that will appear in the delivery report and should be referred to when sending questions to support.
resultCode	Integer	The result code. See Result Codes for more information.
resultDescription	String	This is the textual description for the result code.

5.7 LookupParameters

Name	Data type	Description
username	String	This is the username which is used for authentication. This is provided by Support.
serviceId	Integer	This is the serviceId which is used for authentication. This is provided by Support.
password	String	This is the password which is used for authentication. This is provided by Support.
msisdn	String	This is the MSIDN that should be resolved. Mobile number on international format starting with +.

5.8 LookupResult

Name	Data type	Description
operator	String	This is the resolved operator which the system will use when sending messages to the specified MSISDN. Format is for example: se.telia, no.telenor, dk.tdc
resultCode	Integer	The result code. See Result Codes for more information.
resultDescription	String	This is the textual description for the result code.

5.9 CustomParameter

Name	Data type	Description
key	String	Key parameter. Valid keys and values may be given by support if needed.
value	String	Value parameter.

6 Constants in the WebService

6.1 TON

TON stands for “type of number” and describes how the number should be presented source and destination.

Value (decimal)	Description
0	Short number; 1-5 digits
1	Alphanumeric; Up to 11 valid GSM default alphabet characters. Some operators don’t accept all the characters. Safe characters are A-Z, a-z, 0-9.
2	MSISDN; A mobile number on international format starting with +.

6.2 DCS

DCS stands for “Data Coding Scheme” and describes how the data should be presented. Basic values that are used when sending are:

Value (hex)	Description
00	GSM default alphabet encoding
04	8-bit binary data
08	UCS2 encoded

More information about DCS can be read in the ETSI specification GSM 03.38.

6.3 CustomParameters

Key	Description
chargeOnly	“true” or “false”; indicates that the message only should be charged, not possible for all markets.
async	“true” or “false”; indicates if the message should be sent asynchronous instead of synchronous. Note that premium should be sent as asynchronous.
productDescription	Used for premium SMS. Description on the invoice, market specific.
productCategory	Used for premium SMS. Indicates the category of purchase, contact support for appropriate value.
age	Used for premium SMS. Indicate the required age for the customer, market specific
referenceId	Used for premium SMS. Reference ID from MO SMS, market specific.

7 Result Codes

These are the result codes that may appear in the resultCode variable in SendResult and LookupResult.

Code	Description
0	Unknown error
1	Temporary routing error
2	Permanent routing error
3	Maximum throttling exceeded
4	Timeout
5	Operator unknown error
6	Operator error
100	Service not found
101	User not found
102	Account not found
103	Invalid password
104	Configuration error
105	Internal error (internal Link Mobility error)
106	Quota Exceeded
200	OK (this code will only be used by methods that not deliver messages)
1000	Sent
1001	Delivered
1002	Expired
1003	Deleted
1004	Mobile full
1005	Queued
1006	Not delivered
1007	Delivered charging delayed
1008	Charged message not sent
1009	Charged message not delivered
1010	Expired no delivery report
1011	Billed OK and sent (to operator)
1012	Delayed (temporary billing error, system will try to resend)
1013	Message sent to operator, Bill delayed
2000	Invalid source number
2001	Short number is not supported as source
2002	Alpha is not supported as source
2003	MSISDN is not supported as source number
2100	Short number is not supported as destination
2101	Alpha is not supported as destination
2102	MSISDN is not supported as destination
2103	Operation blocked
2104	Unknown subscriber
2105	Destination blocked
2106	Number error
2107	Destination temporary blocked
2108	Invalid destination
2200	Charging error

2201	Subscriber has low balance
2202	Maximum purchase exceeded
2203	Customer too young
2204	Prepaid subscriber not allowed
2205	Service rejected by subscriber
2206	Subscriber not registered in payment system
2207	Subscriber has reached max balance
2300	Refunded
2301	Could not refund due to illegal or missing MSISDN
2302	Could not refund due to missing messageId
2303	Charged message is queued for refund
3000	GSM encoding is not supported
3001	UCS2 encoding is not supported
3002	Binary encoding is not supported
4000	Delivery report is not supported
4001	Invalid message content
4002	Invalid tariff
4003	Invalid user data
4004	Invalid user data header
4005	Invalid data coding
4006	Invalid VAT
4007	Unsupported content for destination

8 GSM default alphabet

The default GSM alphabet is the alphabet that will be used when specifying DCS 0 (default DCS).

The following characters will use 1 byte per character.

Unicode (hex)	Character
0040	@
00a3	£
0024	$
00a5	¥
00e8	è
00e9	é
00f9	ù
00ec	ì
00f2	ò
00c7	Ç
000a	<LINE FEED>
00d8	Ø
00f8	ø
000d	<CARRIAGE RETURN>
00c5	Å
00e5	å
0394	Δ
005f	_
03a6	Φ
0393	Γ
039b	Λ
03a9	Ω
03a0	Π
03a8	Ψ
03a3	Σ
0398	Θ
039e	Ξ
00c6	Æ
00e6	æ
00df	ß
00c9	É
0020	<SPACE>
0021	!
0022	"
0023	#
00a4	¤
0025	%
0026	&
0027	'

0028	(
0029	)
002a	*
002b	+
002c	,
002d	-
002e	.
002f	/
0030	0
0031	1
0032	2
0033	3
0034	4
0035	5
0036	6
0037	7
0038	8
0039	9
003a	:
003b	;
003c	<
003d	=
003e	>
003f	?
00a1	i
0041	A
0042	B
0043	C
0044	D
0045	E
0046	F
0047	G
0048	H
0049	I
004a	J
004b	K
004c	L
004d	M
004e	N
004f	O
0050	P
0051	Q
0052	R
0053	S
0054	T
0055	U

0056	V
0057	W
0058	X
0059	Y
005a	Z
00c4	Ä
00d6	Ö
00d1	Ñ
00dc	Ü
00a7	§
00bf	¿
0061	a
0062	b
0063	c
0064	d
0065	e
0066	f
0067	g
0068	h
0069	i
006a	j
006b	k
006c	l
006d	m
006e	n
006f	o
0070	p
0071	q
0072	r
0073	s
0074	t
0075	u
0076	v
0077	w
0078	x
0079	y
007a	z
00e4	ä
00f6	ö
00f1	ñ
00fc	ü
00e0	à

8.1 Extended GSM characters

The following characters will be handled as extended characters and will use 2 bytes of space per character.

Unicode (hex)	**Character**
000c	<FORM FEED>
005e	^
007b	{
007d	}
005c	\
005b	[
007e	~
005d	]
007c	|
20ac	€

9 Appendix 1 - Supported TLS versions

To ensure the highest level of security and performance, TLS 1.3 is strongly recommended for all connections to the API. TLS 1.3 offers several advantages over previous versions, including:

- Improved Performance: Faster handshake process, reducing connection latency.
- Stronger Security: Removal of obsolete and vulnerable cryptographic algorithms (e.g., SHA-1, RC4, and static RSA).
- Forward Secrecy: Enhanced protection of session keys, preventing decryption even if the server's private key is compromised.
- Simplified Protocol: Reduced complexity leads to fewer implementation errors and better maintainability.

Although TLS 1.2 is still supported for backward compatibility, it is considered legacy. Clients and servers should be updated to use TLS 1.3 wherever possible.

HTTP is deprecated and LINK strongly recommend using HTTPS if HTTP is being used today.

Supported Ciphers

TLS	Ciphers
1.3	TLS_AES_128_GCM_SHA256 (0x1301) TLS_AES_256_GCM_SHA384 (0x1302) TLS_CHACHA20_POLY1305_SHA256 (0x1303)
1.2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f) TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030) TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca8) Support for the following ciphers below is **removed 2025-10-15**: TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 (0x9e) TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (0x9f) TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xc028) TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xc027) TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013) TLS_RSA_WITH_AES_256_CBC_SHA256 (0x3d) TLS_RSA_WITH_AES_128_CBC_SHA256 (0x3c)