A direct link to your customers

– by the number one provider
of mobile communication solutions

LINK Mobility SMSC Simple HTTP

User guide
Version 5.2; Last updated February 10, 2026

For help, see the following link https://linkmobility.com/support/
The most up-to-date version of this document is available at
http://www.linkmobility.com/developers/

link mobility

Contents

Internal

Before you begin

Please make sure that Link Mobility Support has provided you with the following information:

ServiceId, Username, Password

If you will be receiving Delivery Reports for your messages, please provide Link Mobility Support with an URL. For more information on Delivery Reports, see the "Delivery Reports" chapter. To use Delivery Reports, make sure you have made an opening in any firewalls so that Common can connect to you to transfer Delivery Reports. The addresses to open for are listed below in Appendix 1

Base URL:s

You will get one of these URL:s assigned to you when your account is created:

```
https://n-eu.linkmobility.io/smscsimplehttp
https://c-eu.linkmobility.io/smscsimplehttp
https://s-eu.linkmobility.io/smscsimplehttp
https://se.linkmobility.io/smscsimplehttp
https://de.linkmobility.io/smscsimplehttp
https://es.linkmobility.io/smscsimplehttp
https://wsx.sp247.net/smscsimplehttp (legacy)
http://wsx.sp247.net/smscsimplehttp (legacy)
```

Scope of this document

This document describes the Application Programming Interface (API) for sending text messages through the Link Mobility "Common" platform using query parameters. It also details the mechanism for delivering Delivery Reports to your platform for each message sent. A separate document outlines the API for receiving text messages. The Common API operates over standard HTTP requests, with all interactions performed through URL query parameters. A basic understanding of HTTP and how to construct and interpret query strings is assumed.

Capabilities of "Common" platform

This document describes the Application Programming Interface (API) used to transmit data via query parameters through the designated endpoint. It also outlines the process by which delivery confirmations are returned to your platform for each request submitted. A separate document covers the API for receiving incoming data. This API relies on standard HTTP requests, with all commands and data passed using URL query parameters. It assumes a basic understanding of HTTP protocols and familiarity with constructing query strings

How to send SMS

Perform a POST to any of the Base URLs with the following query parameters

Query parameters

Query parameter	Data type	Description
ServiceId	Integer	This is the serviceId which is used for authentication. This is provided by Support.
Username	String	This is the username 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. Usage allows full international numbers (+46732594084), short numbers (72401) or text (LINK).
Destination	String	This is the destination number.
Userdata	String	This is the message itself. 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.

		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.
UserdataHeader	String	This value may be specified when sending concatenated SMS. More information about valid UDH for long SMS may be given by support upon request.
ReferenceId	String	For premium SMS. Use id of incoming SMS here.
Tariff	String	For premium SMS. To charge 10 SEK use tariff SEK1000. Set to null or SEK0 for bulk messages.
Vat	Double	Vat for premium SMS. 0.0 means no Vat, 25.0 a Vat of 25%.
ChargeOnly	String	Set to true for a silent charging without any SMS sent, otherwise false.
Async	String	True for asynchronous call, initial status will be Queued (1005) for a successful call.

Returns

Response will contain a message with a result code, result description and a message id if successful separated with ";", i.e. *resultCode;resultDescription;messageId*.

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

Delivery Reports

When a mobile-terminated (MT) message is delivered to a handset—or if delivery fails for any reason—a delivery report will be sent to your system via callback. Delivery reports are mandatory for charged messages and optional (though strongly recommended) for free messages. For multi-part messages, a separate delivery report will be sent for each part.

Reports can be delivered in JSON, XML, or as HTTP GET/POST key-value pairs. If you need to change the format or callback URL, please contact Support.

Your receiver must respond with an HTTP 200 OK status to confirm successful receipt of the delivery report. Optionally, for added reliability, your receiver can also be configured to return a specific string in the response body. If you wish to enable this feature, please contact Support to have it configured on your Gateway.
Delivery reports will be sent via HTTP POST from specific IP addresses. Ensure that your firewall allows traffic from the IPs listed in Appendix 1 of this document.

Delivery reports contain the following fields:

Field	Data type	Description
refId	String	If you used a refId when submitting the message, this will be mirrored here. If not, this will be null.
id	String	This is Common's internal message ID for this message. It mirrors the ID which was given

		to you when submitting the message. If the message is a multipart message, the id will have the following pattern *{id}${n}** where *n* is the ordinal number that identifies the part. ***Example:** *abc123$0* is the first part of the message with id *abc123*
operator	String	The telecom operator the message was sent to (The end-users's operator)
sentTimestamp	DateTime	The timestamp when Common sent the message to the telecom operator. UTC time formatted according to RFC3339.
timestamp	DateTime	The timestamp from the telecom operator for this status message. UTC time formatted according to RFC3339.
resultCode	Integer	The status of the message. For what the different codes mean, see Status codes table below.
operatorResultCode	String	The unmapped status of the message from the operator. Each telecom operator has different statuses and this is only provided for debugging or reference, resultCode is the real status.
segments	Integer	The number of segments (of 140 bytes) the message was split into for delivery.
gateCustomParameters	<List>KeyValue	If there are any custom parameters set on your gate, they will be provided here. Usually blank.
customParameters	<List>KeyValue	If there are any extra fields in the delivery report Common receives from the operator, they will be listed here. Note: The parameters source and destination (as defined in the request) added by default to the customParameters.

SMSC Simple HTTP – User guide

Delivery Report Example

The following example is an example of a successfully delivered message. refId and id have been set to invalid values in this example.

```
{
  "refId": "0",
  "id": "0",
  "operator": "no.telenor",
  "sentTimestamp": "2015-11-19T09:37:35Z",
  "timestamp": "2015-11-19T09:37:00Z",
  "resultCode": 1001,
  "operatorResultCode": "2",
  "segments": 1,
  "gateCustomParameters": {},
  "customParameters": {
    "received": "2015-11-19 10:37:36",
    "source": "2333",
    "destination": ""+4746910822"
  }
}
```

The following example is an example of a message which was attempted sent to a phone number which does not exist. refId and id have again been set to invalid values in this example.

```
{
  "refId": "0",
  "id": "0",
  "operator": null,
  "sentTimestamp": "2015-11-19T10:17:37Z",
  "timestamp": "2015-11-19T10:17:37Z",
  "resultCode": 2106,
  "operatorResultCode": null,
  "segments": 1,
  "gateCustomParameters": {},
  "customParameters": {
    "received": "2015-11-19 11:17:37",
    "source": "2333",
    "destination": ""+4746910823"

  }
}
```

Result Codes

The most common result code is 1001 Delivered. This code indicates a successful delivery (and payment, if charged) of the message. Most statuses are final, indicating that the message either has been successfully delivered, or failed in a non-recoverable way.

resultCode	Description	Transaction State
0	Unknown error	FINAL: NOT DELIVERED, NOT BILLED*
1	Temporary routing error	FINAL: NOT DELIVERED, NOT BILLED*
2	Permanent routing error	FINAL: NOT DELIVERED, NOT BILLED*
3	Maximum throttling exceeded	FINAL: NOT DELIVERED, NOT BILLED*
4	Timeout	FINAL: UNKNOWN DELIVERY, UNKNOWN BILLING*
5	Operator unknown error	FINAL: UNKNOWN DELIVERY, UNKNOWN BILLING*
6	Operator error	FINAL: NOT DELIVERED, NOT BILLED*
104	Configuration error	FINAL: NOT DELIVERED, NOT BILLED*
105	Internal error (internal Link Mobility error)	FINAL: NOT DELIVERED, NOT BILLED*
106	Quota Exceeded	FINAL: NOT DELIVERED, NOT BILLED*
1000	Sent (to operator)	**TEMP**: NOT DELIVERED, NOT BILLED*
1001	Billed and delivered	FINAL: DELIVERED, BILLED* (if applicable)
1002	Expired	FINAL: NOT DELIVERED, NOT BILLED*
1003	Deleted	FINAL: NOT DELIVERED, NOT BILLED*
1004	Mobile full	FINAL: NOT DELIVERED, NOT BILLED*
1005	Queued	**TEMP**: NOT DELIVERED, NOT BILLED*
1006	Not delivered	FINAL: NOT DELIVERED, NOT BILLED*
1007	Delivered, Billed delayed	**TEMP**: DELIVERED, NOT BILLED*
1008	Billed OK (charged OK before sending message)	**TEMP**: NOT DELIVERED, BILLED*

1009	Billed OK and NOT Delivered	FINAL: NOT DELIVERED, BILLED*
1010	Expired, generated by LINK	FINAL: UNKOWN DELIVERY, UNKNOWN BILLING*
1011	Billed OK and sent (to operator)	**TEMP**: NOT DELIVERED, BILLED*
1012	Delayed (temporary billing error, system will try to resend)	**TEMP**: NOT DELIVERED, NOT BILLED* (resending)
1013	Message sent to operator, Bill delayed	**TEMP**: NOT DELIVERED, NOT BILLED*
2000	Invalid source, the specified source number or Alpha is invalid	FINAL: NOT DELIVERED, NOT BILLED*
2001	Source shortnumber not supported, the source TON may not be set to shortnumber	FINAL: NOT DELIVERED, NOT BILLED*
2002	Source alpha not supported, the source TON may not be set to alpha	FINAL: NOT DELIVERED, NOT BILLED*
2003	Source MSISDN not supported, the source TON may not be set to MSISD	FINAL: NOT DELIVERED, NOT BILLED*
2100	Destination shortnumber not supported, the destination TON may not be set to shortnumber	FINAL: NOT DELIVERED, NOT BILLED*
2101	Destination alpha not supported, the destination TON may not be set to alpha	FINAL: NOT DELIVERED, NOT BILLED*
2102	Destination MSIDN not supported, the destination TON may not be set to MSISDN	FINAL: NOT DELIVERED, NOT BILLED*
2103	Operation blocked, requested operation is not supported for the specified destination	NOT DELIVERED, NOT BILLED*
2104	Unknown subscriber	FINAL: NOT DELIVERED, NOT BILLED*
2105	Destination blocked (subscriber permanently barred)	FINAL: NOT DELIVERED, NOT BILLED*
2106	Number error	FINAL: NOT DELIVERED, NOT BILLED*
2107	Destination temporarily blocked (subscriber temporarily barred)	FINAL: NOT DELIVERED, NOT BILLED*
2108	Invalid destination	FINAL: NOT DELIVERED, NOT BILLED*

Internal

2200	Charging error	FINAL: NOT DELIVERED, NOT BILLED*
2201	Subscriber has low balance	FINAL: NOT DELIVERED, NOT BILLED*
2202	Subscriber barred for overcharged (premium) messages	FINAL: NOT DELIVERED, NOT BILLED*
2203	Subscriber too young (for this particular content)	FINAL: NOT DELIVERED, NOT BILLED*
2204	Prepaid subscriber not allowed	FINAL: NOT DELIVERED, NOT BILLED*
2205	Service rejected by subscriber	FINAL: NOT DELIVERED, NOT BILLED*
2206	Subscriber not registered in payment system	FINAL: NOT DELIVERED, NOT BILLED*
2207	Subscriber has reached max balance	FINAL: NOT DELIVERED, NOT BILLED*
3000	GSM encoding is not supported	FINAL: NOT DELIVERED, NOT BILLED*
3001	UCS2 encoding is not supported	FINAL: NOT DELIVERED, NOT BILLED*
3002	Binary encoding is not supported	FINAL: NOT DELIVERED, NOT BILLED*
4000	Delivery report is not supported	FINAL: NOT DELIVERED, NOT BILLED*
4001	Invalid message content	FINAL: NOT DELIVERED, NOT BILLED*
4002	Invalid tariff	FINAL: NOT DELIVERED, NOT BILLED*
4003	Invalid user data	FINAL: NOT DELIVERED, NOT BILLED*
4004	Invalid user data header	FINAL: NOT DELIVERED, NOT BILLED*
4005	Invalid data coding	FINAL: NOT DELIVERED, NOT BILLED*
4006	Invalid VAT	FINAL: NOT DELIVERED, NOT BILLED*
4007	Unsupported content for destination	FINAL: NOT DELIVERED, NOT BILLED*

Appendix 1

The following hosts are currently used for outgoing messaging.

Hostname(s)	IP address(es)
sp1.n-eu.linkmobility.io	193.142.108.11
sp2.n-eu.linkmobility.io	193.142.108.12
sp3.n-eu.linkmobility.io	193.142.108.13
sp4.n-eu.linkmobility.io	193.142.108.14
sp5.n-eu.linkmobility.io	193.142.108.15
sp6.n-eu.linkmobility.io	193.142.108.16
sp1.c-eu.linkmobility.io	37.202.24.11
sp2.c-eu.linkmobility.io	37.202.24.12
sp3.c-eu.linkmobility.io	37.202.24.13
sp4.c-eu.linkmobility.io	37.202.24.14
sp5.c-eu.linkmobility.io	37.202.24.15
sp6.c-eu.linkmobility.io	37.202.24.16
sp1.s-eu.linkmobility.io	185.73.14.11
sp2.s-eu.linkmobility.io	185.73.14.12
sp3.s-eu.linkmobility.io	185.73.14.13
sp4.s-eu.linkmobility.io	185.73.14.14
sp5.s-eu.linkmobility.io	185.73.14.15
sp6.s-eu.linkmobility.io	185.73.14.16
socks1.sp247.net	195.84.162.34
socks2.sp247.net	194.71.165.71
socks3.sp247.net	195.84.162.16
socks4.sp247.net	194.71.165.98
socks5.sp247.net	195.84.162.3
socks6.sp247.net	194.71.165.122
s1.n-eu.linkmobility.io	213.242.87.36
s2.n-eu.linkmobility.io	213.242.87.37
s3.n-eu.linkmobility.io	213.242.87.38
s4.n-eu.linkmobility.io	213.242.87.39
s5.n-eu.linkmobility.io	213.242.87.40
s6.n-eu.linkmobility.io	213.242.87.41
s1.c-eu.linkmobility.io	62.67.62.101
s2.c-eu.linkmobility.io	62.67.62.102
s3.c-eu.linkmobility.io	62.67.62.103
s4.c-eu.linkmobility.io	62.67.62.104
s5.c-eu.linkmobility.io	62.67.62.105
s6.c-eu.linkmobility.io	62.67.62.106
s1.s-eu.linkmobility.io	217.163.95.196
s2.s-eu.linkmobility.io	217.163.95.197
s3.s-eu.linkmobility.io	217.163.95.198
s4.s-eu.linkmobility.io	217.163.95.199
s5.s-eu.linkmobility.io	217.163.95.200
s6.s-eu.linkmobility.io	217.163.95.201

Internal

Internal

Appendix 2

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)

Changelog of this document

Date	Version	Author	Changes
2011-04-18	1.0	LH	Created
2013-04-05	2.0	KCN	Updated support for long SMS Added error codes Changed logo
2014-09-18	3.0	JA	Added refund method Added error codes
2020-20-20	4.0	HA	Removed refund method
2025-06-05	5.0	EP	Moved the guide from the old format to the new one
2025-07-03	5.1	KCN	Updated Appendix 2 with TLS changes.
2026-02-10	5.2	FS	Updated IPs