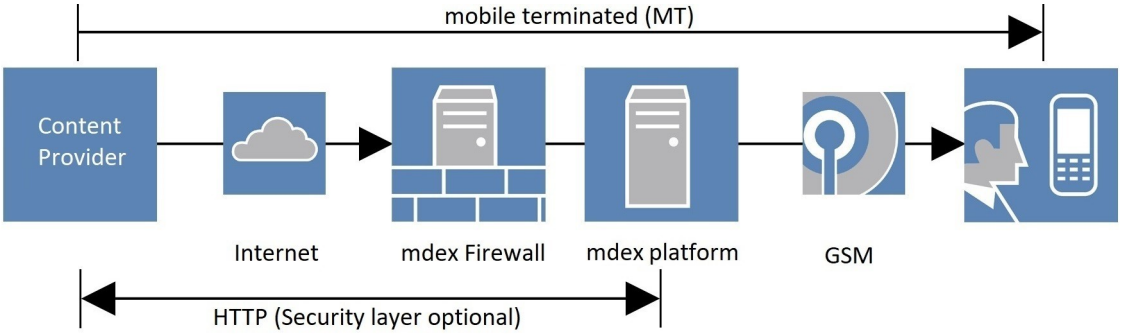
mdex SMS messaging platform – Technical Specification
Protocol: HTTP

Overview

The industry standard protocol "HTTP" is used to

- transmit SMS messages to GSM mobile network devices worldwide.



Standard features

Transmission:	Mobile terminated SMS, (text-messages worldwide and binary content as far as supported by network operators)
Performance:	5 SMS/sec typical performance
Availability:	More than 99% of all cases on an annual base
Client identification:	Client IP address, Customer ID and password
Security layer	≥ TLS1.2 (optional)

Additional optional features

Selected routing:	Selected routing of mobile terminated SMS though hosted large account (LA hosting)
Converting:	Converting mobile terminated SMS to different media (Speech, Email, Fax)
DLR:	Delivery reports supported (if feature enabled; Supported protocols for DLR's: EMI/UCP, SMPP or SMTP)

Format description

To:	http(s)://sm-http.ic3s.de/cgi-bin/sm-http.pl
Method:	GET or POST
Required parameters:	<id> <pw> <adc> <msg>
Optional parameters	<oadc> <alnumoadc> <hex> <mcl> <udh> <dcs> <simtool> <notificationto> <sendstartts> <discardmsg>

Wireless Logic mdex GmbH
Bäckerbarg 6; D-22889 Tangstedt
Phone +49 (0)4109 555-0
email sales@mdex.de
Fax +49 (0)4109 555-57
Web www.mdex.de

Process description

The following resources are needed at the customer site:

- Fixed internet access with one or more IP addresses
- Client software for this protocol

The following information are needed from the customer:

- Activation sheet for this protocol, including customer name, technical contact, email address for maintenance information and sender IP address(es) of your client

The following information is communicated to the customer:

- Customer ID and password
- Connection parameters
- Certificates for verification purposes (optional for security layer)
- Access data to customer lounge (optional)

Glossar

Required parameters:

<id>	Communicated Customer ID, max 32 chars 7bit ASCII
<pw>	Communicated password, max 32 chars 7bit ASCII
<adc>	<u>Send SMS to one recipient:</u> Phone number of the recipient (AdC) Format "+491721234567", "00491721234567" or for German recipients "01721234567". <u>Send same SMS message text to multiple recipients:</u> Multiple recipients can be specified by separating them with the comma sign (,), eg. adc=+491721234567,+491731234567,+491741234567,+491751234567 Use the HTTP POST method for huge amount of recipients. Each SMS will be sent out and billed to your account the same way as if they were submitted by individual HTTP requests. A separate application return code will be returned for each generated SMS (one per line) in same order as AdC's specified. Invalid AdC will be skipped and flagged with the appropriate return code. If message was auto-splitted (see below), the number of separate application return code for each AdC depends on the number of parts generated for a concatenated SMS by the auto-split function.
<msg>	Message: By default coded as readable ASCII text (>20 hex), acc. to ISO 8859-1 See also "Special characters within message text" Maximum 160 chars for one standard text SMS Auto-split function to a concatenated standard text SMS if <msg> parameter exceeds 160 chars. Default: Auto-split to a max. four part concatenated SMS (maximum optional adjustable) Max. chars in <msg> for a four part concatenated SMS: 4 x 153 chars = 612 chars. Each SMS part of the concatenated SMS will be sent out separately and billed to your account. If "hex" flag is used: Coded as hexadecimal string, max. 280 chars, spaces are ignored. A separate application return code will be returned for each generated SMS (one per line)

Optional parameters:

<oadc>	Phone number of originator (OadC) Format "+491721234567", "00491721234567" or, for a German OAdC "01721234567"
<alnumoadc>	Alphanumeric originator: Replaces any numeric OAdC Allowed characters a->z,A->Z,0->9,@ÄÖÜäöüß, max. 11 chars Please note: Spaces must be embedded within quotation marks Character encoding according to RFC 2068 HTTP 1.1
<hex>	The <message> field is coded in hexadecimal code
<mcl>	Supported message classes, See GSM 03.38 0: direct display; 1: ME-specific, e.g. ring tones etc.; 2: SIM specific; 3: TE-specific.
<udh>	The "user data header"; See GSM 03.40, 9.2.3.23
<dcs>	Data coding scheme, See GSM 03.38 00: default 04: 8 bit char set, typ. Latin-1 08: UCS-2 char
<simtool>	The "Sim Toolkit" bit is set. See GSM 03.40, 9.2.3.9
<notificationto>	Email address for requested delivery status (if feature enabled)
<sendstartts>	Deferred SMS delivery – Start delivery of SMS at sendstartts Supported timestamps: ISO 8601 - Combined date and time representations with T – delimiter Timestamp examples: YYYY-MM-DDThh:mm:ss → German local time YYYYMMDDThhmmssZ → UTC timestamp YYYY-MM-DDThh:mm:ss[+-]hh:mm → Timezone declaration
<discardmsg>	If set to "1" the generated SMS(es) will not be sent out to addressed recipient. Ability to check e.g. application return code at no cost for SMS delivery

Special characters within message text

Special chars:	<u>German Umlauts</u> Please note: Submit these chars as hex value as they are not supported as ASCII text, acc. to ISO 8859-1 Chars: ä → %e4 ö → %f6 ü → %fc Ä → %c4 Ö → %d6 Ü → %dc <u>Chars from GSM 03.38 - Basic Character Set Extension</u> Please note: Escape code required for these chars, using two chars within the GSM 7-bit encoded message Not all chars supported by MNO's and/or mobile phones Chars: ^ → %1B%14 { → %1B%28 } → %1B%29 \ → %1B%2F [→ %1B%3C ~ → %1B%3D] → %1B%3E | → %1B%40 € → %1B%65 <u>Examples of further chars supported by GSM 03.38</u> Please note: Any char of the GSM 7-bit default alphabet can be submitted as its hex value: Chars: Line Feed → %0A Ω → %15 … ***<u>Caution: Avoid double encoding of hexadecimal represented characters</u>*** Please note: ä → %e4 has to stay %e4 and should not be encoded into %25e4

Methode GET – Examples:

Base URL: "*http://sm-http.ic3s.de/cgi-bin/sm-http.pl?id=id&pw=pw*", followed e.g. by below mentioned parameters.
Replace id with your customerID and pw with your password.
Note: All parameter values must be set URL-encoded. See RFC2396 for reserved characters.

simple SMS:	&adc=%2B491727654321&msg=text
simple SMS with alnumOAdC:	&adc=%2B491727654321&alnumoadc=Company&msg=text
Direct display text-message	&adc=%2B491727654321&mcl=0&msg=text
simple SMS with umlauts:	&adc=%2B491717654321&msg=%E4%F6%FC%DF
complex message (e.g. operator logo for Nokia mobiles):	&adc= %2B491717654321&hex=1&udh=1&mcl=1&msg=0605041582000062F22000480E010007FE0000000000 000004390FC1C00F80000004D08787E01FC00000059083066018C000000510830C2018000000070903 0C00180000000606030C000F0000000602030C03078000000701030C0480C000000500830C2090C000 00058083066118C0000004C08787E09FC0000004310FC1C48F80000007FE000003000000
SMS with UCS2 charset	&adc=%2B491717654321&hex=1&dcs=08&msg=0054006500730074
Simple SMS with several recipients:	&msg=text&adc=+491721234567,+491731234567,+491741234567,+491751234567

Example with curl get request
Message text: "This is a http get message with Linefeed LF, Umlauts ä ö ü and Eurosign €"
curl -X GET -k -i 'http://sm-http.ic3s.de/cgi-bin/sm-http.pl?id=id&pw=pw&adc=%2B491727654321&msg=This%20is%20a%20http%20get%20message%20with%20Linefeed,%0AUmlauts%20%e4%20%f6%20%fc%20and%20Eurosign%20%1B%65.'

Methode POST – Examples:

Curl post-example:
Message text: "This is a http post message with Linefeed LF, Umlauts ä ö ü and Eurosign €"
curl -X POST https://sm-http.ic3s.de/cgi-bin/sm-http.pl -d 'id=id&pw=pw&adc=%2B491727654321&msg=This is a http post message with Linefeed,%0AUmlauts %e4 %f6 %fc and Eurosign %1B%65.'

HTML form example with post method and action URL.
Label input type text name attribute to parameter name-

```
<html>
  <body>
    <form action="http(s)://sm-http.ic3s.de/cgi-bin/sm-http.pl" method="post">
    <input type="text" name="id">
    <input type="text" name="pw">
    <input type="text" name="adc">
    <input type="text" name="msg">
    <input type="submit" name="Sent" value="Sent">
    </form>
  </body>
</html>
```

Important note:

As parameters are not visible in the address field of your browsers, we suggest using the POST method. Please keep in mind, that you should not use your id and/or password in any file that users are able to access. We cannot be held responsible for any misuse of your id/password – it is your task to keep it secure.

HTTP status codes:

See RFC 2616 - Status Code Definitions

Application return codes

After sending a message following application return information will be received within the HTTP message body.
Application return code followed by a short explanation (in one line) for each generated SMS and one or more additional details, separated by " --".

xxx = replaced by individual text

200 >> OK jobid >> LH:SMHTTP@xxx:messageID
204 >> empty message
400 >> ADC empty
400 >> ADC wrong (xxx)
400 >> invalid ADC (adc=xxx)
400 >> binary message too long for xxx
401 >> empty ID
401 >> bad authentication
401 >> empty password
401 >> bad authentication
401 >> unauthorized IP (xxx)
405 >> method not allowed
406 >> bad message class
406 >> bad HEX message xxx
407 >> Invalid timestamp format xxx
500 >> configuration file not found
500 >> cannot create xxx
500 >> cannot rename xxx
503 >> IP no longer valid
503 >> password no longer valid
503 >> xxx is not allowed to use this service
503 >> no SIM-Toolkit Message allowed

Examples of application return codes with one or more additional details, separated by " -- "

200 >> OK jobid >> LH:SMHTTP@xxx:messageID -- MESSAGE DISCARDED
200 >> OK jobid >> LH:SMHTTP@xxx:messageID -- Immediate delivery, sendstartts=<TS> in the past
200 >> OK jobid >> LH:SMHTTP@xxx:messageID -- Delivery postponed to <TS> – MESSAGE DISCARDED