

HTTP Access

Description



| | |
|------------------|--|
| Filename | F24-Schweiz_Beschreibung_HTTP-HTTPS-2.21_EN.docx |
| Version | 2.21 |
| Last change date | 14.01.2025 |
| Document Owner | F24 Schweiz AG, Wollerau, Switzerland |
| Classification | public |

Contents

| | |
|---|----|
| 1. Introduction..... | 4 |
| 2. Requirements | 4 |
| 3. Message Formats..... | 4 |
| 3.1. Sending text messages | 5 |
| 3.2. Sending fax messages | 8 |
| 3.3. Sending voice messages..... | 10 |
| 3.4. Status query..... | 12 |
| 3.4.1. Status query for a particular message..... | 12 |
| 3.4.2. Status query for a space of time | 12 |
| 3.4.3. Answer structure for a state query | 14 |
| 3.4.3.1. Status codes text messages | 15 |
| 3.4.3.2. Status codes voice messages..... | 17 |
| 3.4.3.3. Status codes fax messages | 18 |
| 3.5. Message Buffer..... | 20 |
| 3.5.1. Query for new messages | 20 |
| 3.5.2. Delete messages from the buffer..... | 22 |
| 3.6. Receiving replies to an SMS | 23 |
| 3.7. Notification message | 24 |
| 3.7.1. Notification result codes..... | 25 |
| 3.8. Forwarding received SMS messages to HTTP | 26 |
| 3.8.1. Forwarding via GET requests | 26 |
| 3.8.2. Forwarding via POST requests..... | 26 |
| 4. Reply codes..... | 28 |
| 4.1. HTTP status codes | 28 |
| 4.2. eCall ResultCodes | 29 |
| 5. Connections | 31 |
| 5.1. Access addresses for sending jobs | 31 |

| | | |
|------|---|----|
| 5.2. | IP addresses for replies or notifications (eCall to client) | 31 |
| 6. | Header information..... | 32 |
| 6.1. | Client Header ID | 32 |
| 6.2. | Server Header ID..... | 32 |
| 7. | Access test..... | 33 |
| 8. | Table directory..... | 34 |

1. Introduction

This document describes eCall HTTP/HTTPS access. eCall's HTTP access allows you to open a URL directly, in order to send messages to various call systems.

2. Requirements

To allow you to send messages via the HTTP interface, the following conditions must be met:

- You must have a business account with eCall (business package). A standard account (private package) is adequate for testing (30 days)
- You must activate the HTTP interface in your eCall account
- Your software must be able to open a URL to send messages.

3. Message Formats

Messages can be sent using GET or POST. The following rules must be followed here:

- All parameters consist of a command word (ID) and the associated, desired value.
- Message elements are separated from each other by the "&" character (compulsory for GET, also for POST where the content type involves "application/x-www-form-urlencoded")
- ID and value are separated by an equals sign "="
- The ID's are case-insensitive (i.e. upper/lower case does not matter)
- All characters must be transferred using UTF8 encoding. In addition, the characters must be URL encoded (e.g. "%20" or "+" for a space).

Example:

<https://url.ecall.ch/api/sms?username=maxmuster&password=s3cr3t&address=0041791234567&message=Dies%20ist%20ein%20Test>

Note: For performance reasons, account settings parameters are cached by the HTTP interface. Hence, changes in the settings may only take effect in the HTTP interface with a certain delay.

3.1. Sending text messages

- **Function**

Api/Sms Function for a text message:

- **Required Parameters**

UserName Account username or self-created API credentials (*Interfaces > Overview*)

Password User password.

Address Complete pager or mobile number (in international format), or several numbers, separated by a semicolon, or the name of a person or group which is defined in the eCall account address book.

Note: When the first invalid address is found, the checking process and further processing will be cancelled.

Message Text which is to be sent.
The maximum length is determined in the account settings (*Account Settings > SMS > Maximum number of pages*). Text which is longer than allowed by the settings will be truncated.

- **Optional Parameters**

JobID Unique message identification indicated by the client.
A maximum length of 50 characters will be accepted. If the value is longer, it will be truncated.

SendDate If the job Send Date and Time defines that it should be sent later, it should be defined in ISO 8601 format (e.g. in the format "YYYY-MM-DDThh:mm:ss±hh:mm")

Note: If a time zone is not indicated, eCall will use the local time zone. UTC can be indicated with "Z" as well as "+00" or "+00:00".

Example for time zone UTC+3: 2019-08-01T10:09:00+03:00

Example for time zone UTC: 2019-08-01T10:09:00Z

- **Extended functions**

NotificationAddress Address for forwarding a notification (e.g. confirmation of receipt). Maximum length is 100 characters.

Possible values:

- Mobile number (e.g.: 0041791234567)
- Email address (e.g. john.template@example.com)
- URL (e.g.: https://www.example.com/notifcations.asp)

Values may also be combined, where the individual values are separated by a semicolon “;” as long as the maximum length is not exceeded e.g.:

0041791234567;0041781234567;john.template@example.com

Note: The validation will fail at the first invalid address found, and processing of the request is aborted.

Under [Receiving replies to an SMS](#) there is a description of what a reply message via URL looks like

NotificationLevel Value indicating when a confirmation of receipt is desired.

Possible values:

- 0 – (Default) Confirmation of receipt only when a receipt status is present.
- 1 – Confirmation of receipt once the last possible monitoring point has been reached.
- 2 – Send confirmation of receipt as per Value 1, and when the job is still not sent after a number of seconds*.
* duration determined by the telecom provider.
- 3 – Send confirmation of receipt when the job could not be sent, i.e. in the case of sending errors or timeout when contacting the end device.

CallBack Sender, up to 16 numeric or up to 11 alphanumeric characters.

Answer Recipient addresses for forwarding an SMS response can be defined here.

Possible values: (see *Parameter NotificationAddress*)

Note: If the “Answer” parameter is also sent, the “CallBack” value is ignored.

MsgType Transfer possibility for a certain information type.

Valid types:

- Normal – Default
- Flash – Flash-SMS

- PrioSMS+ – a Flash and a “Normal” SMS

NoLog With this parameter you can prevent messages from showing up in the log book. Valid values are “1” or “0” (default).

- **Examples**

```
/Api/Sms?Address=%2B41791112233&Message=Hello%20World&Username=xxx&Password=yyy
```

```
/Api/Sms?Address=%2B41791112233&Message=Hello%20World&Username=xxx&Password=yyy&CallBack=%2B41791234567
```

```
/Api/Sms?Address=%2B41791112233&Message=Hello%20World&Username=xxx&Password=yyy&MsgType=Flash&NoLog=1
```

Note: At most 1530 characters (GSM encoding, normal case), respective 1570 characters (UCS-2 encoding) are allowed per message. As there are only 160 characters available in a SMS, it may be necessary to split the message into several partial messages (pages). At most 10 pages are allowed. For the receiver’s device to be able to re-assemble these partial messages into the original message, the respective data is included in each page. This reduces the number of characters available per page for the actual message by 7 (GSM encoding), respectively by 3 (UCS-2 encoding).

3.2. Sending fax messages

- **Function**

Api/Fax Function for a fax message.

- **Required Parameters**

UserName Account username or self-created API credentials (*Interfaces > Overview*)

Password User password.

Address Complete fax number (in international format), or several numbers, separated by a semicolon, or name of a person or group, which is defined in the address book of the eCall account.

Message Fax message text

- **Optional Parameters**

JobID Unique message identification indicated by the client.
A maximum length of 50 characters will be accepted. If the value is longer, it will be truncated.

SendDate If the job is to be sent in the future this parameter defines the date and time it should be sent in ISO 8601 format (e.g. in the format "YYYY-MM-DDThh:mm:ss±hh:mm")
Note: If a time zone is not indicated, eCall will use the local time zone. UTC can be indicated with "Z" as well as "+00" or "+00:00".
Example for time zone UTC+3: 2019-08-01T10:09:00+03:00
Example for time zone UTC: 2019-08-01T10:09:00Z

- **Extended functions**

CallingNumber Sender number (international format)

Subject Title of the message. Added on the coversheet.

HeaderId Sender's header. Added at the top of each page.

HeaderInfo Header for info. Added beside the HeaderId on each page.

NumberOfRetries Maximum number of repeated attempts. Default value is 0, there is only one attempt to execute the job.

RetryIntervallInMinutes Time between retries in minutes.

- **Examples**

`/Api/Fax?Address=%2B41447873071&Message=Test+message&Username=xxx&Password=yyy`

`/Api/Fax?Address=%2B41447873071&Subject=Test&Message=Test+message&Username=xxx&Password=yyy&FromText=%2B41431234576&SendDate=2019-12-20T20:50:00&JobID=123AB`

3.3. Sending voice messages

- **Function**

Api/Voice Function for a voice message.

- **Required Parameters**

UserName Account username or self-created API credentials (*Interfaces > Overview*)

Password User password.

Address Complete recipient number (in international format).

Message Voice message text

- **Optional Parameters**

JobID Unique message identification indicated by the client.
A maximum length of 50 characters will be accepted. If the value is longer, it will be truncated.

SendDate If the job Send Date and Time defines that it should be sent later, it should be defined in ISO 8601 format (e.g. in the format "YYYY-MM-DDThh:mm:ss±hh:mm")
Note: If a time zone is not indicated, eCall will use the local time zone. UTC can be indicated with "Z" as well as "+00" or "+00:00".
Example for time zone UTC+3: 2019-08-01T10:09:00+03:00
Example for time zone UTC: 2019-08-01T10:09:00Z

- **Extended Functions**

Language Defines the language in which the text will be read. The parameters for language are as follows:

- **DE** for German
- **FR** for French
- **IT** for Italian
- **EN** for English

Default: Language will be identified automatically using content.

FromText Name of client to be read after the message.

- **Examples**

/Api/Voice?Address=%2B41447873070&Message=Testmitteilung&Username=xxx&Password=yyy

/Api/Voice?Address=%2B41447873070&FromText=Hans%20Muster&Message=Testmitteilung&Username=xxx&Password=yyy&Language=FR&SendDate=2019-12-20T20:50:00&JobID=123

3.4. Status query

Besides the possibility to be informed of the status of a message via notification (parameter "NotificationAddress" in the send request) it is also possible to query the status of a specific message (by "JobID") or of all messages within a specified time interval.

3.4.1. Status query for a particular message

- **Function**

Api/State Function for a query of the current status.

- **Required Parameters**

UserName Account username or self-created API credentials (*Interfaces > Overview*).

Password User password.

JobID Unique message identification of the original job as indicated by the client.

- **Example**

/Api/State?Username=xxx&Password=yyy&JobID=1234ABCD

3.4.2. Status query for a space of time

- **Function**

Api/State Function for a query of the current status.

- **Required Parameters**

UserName Account username or self-created API credentials (*Interfaces > Overview*).

Password User password.

StartDate Start date and time of the **original jobs** for which the status shall be returned.

EndDate End date and time of the **original jobs** for which the status shall be returned.

Note about StartDate and EndDate: If a time zone is not indicated, eCall will use the local time zone. UTC can be indicated with "Z" as well as "+00" or "+00:00".

Example for time zone UTC+3: 2019-08-01T10:09:00+03:00

Example for time zone UTC: 2019-08-01T10:09:00Z

- **Example**

/Api/State?Username=xxx&Password=yyy&StartDate=2019-12-01T10:00:00&EndDate=2019-12-01T13:00:00

Note: If the number of status objects found for a [start date, end date] interval exceeds 1000, HTTP status code «bad request» (HTTP status codes) and eCall result code **11701** are returned (

eCall ResultCodes).

3.4.3. Answer structure for a state query

The structure is in XML format. For several statuses ([Status query for a space of time](#)), several <Status> elements are grouped below the <Statuses> root element.

Structure for a query (content omitted)

```

<Statuses>
  <Status>
    <StatusCode></StatusCode >
    <StatusText></StatusText >
    <TimeStamp></TimeStamp>
    <Address></Address>
    <JobID></JobID>
  </Status>
  <Status>
    ....
  </Status>
</Statuses>

```

The value of «StatusCode» is determined according to the following tables:

3.4.3.1. Status codes text messages

| SMS / Pager | | |
|-------------|-------------------------------|--|
| Status Code | Status Text | Description |
| -1 | Unkown | Status unknown. |
| 99 | Scheduled | Scheduled transmission time. |
| 0 | Transmitting | Transmitting. |
| 1 | Ok | Transmission OK. |
| 3 | Waiting for result | Transmitting. Waiting for status response. |
| 8 | Ok (confirmed) | Transmission OK (reception confirmed). |
| 9 | Ok (waiting for confirmation) | Transmission OK (reception not yet confirmed, waiting to deliver message). |
| 10 | Ok (not confirmed) | Transmission OK (reception of confirmation failed). |
| 71 | Not accepted by gateway | No «job accepted» message from gateway. |
| 1000 | Disconnected | Connection cancelled. |
| 1001 | Timeout | Timeout: The message could not be sent. |
| 3020 | Connection refused | The connection was refused by the remote host. |
| 5111 | Text too long | The text is too long. |
| 5112 | Invalid number | Invalid number. |
| 5120 | Message invalid | Destination call number is a tone call number: Message invalid. |
| 5121 | Invalid char in number | Destination call number is invalid: Invalid character. |

| SMS / Pager | | |
|--------------------|---|---|
| Status Code | Status Text | Description |
| 5155 | Too much traffic | Too much traffic. |
| 5164 | Call system troubled | Call system troubled. |
| 5502 | Syntax error | Message from provider: Syntax error. |
| 5504 | Too many messages | Message from provider: Operation not allowed - Maximum messages for the address exceeded. |
| 5506 | ADC invalid | Message from provider: Analog-to-digital converter invalid. |
| 5509 | Invalid number | Invalid call number. |
| 5510 | Too many messages for same receiver | Too many messages for same receiver from this account. |
| 5511 | Too many messages with same content for same receiver | Too many messages with same content for same receiver from this account. |
| 5512 | Duplicate job id | This job ID has already been used by this sender. |
| 5513 | Exists in the blacklist | Callback, address or parts of the content are blacklisted |
| 5514 | Blocked by out-bound filter rule | Blocked by out-bound filter rules |

– Table: Status Codes SMS / Pager

3.4.3.2. Status codes voice messages

| Voice | | |
|-------------|----------------------|---|
| Status Code | Status Text | Description |
| 30 | Transmitting | Message is being transferred to voice system. |
| 31 | Ok | Message successfully transferred to voice system. |
| 32 | Failed | Transfer of message to voice system was not successful. |
| 33 | Played | Message played successfully. |
| 34 | Played with feedback | Message played successfully and feedback received. |
| 35 | Invalid number | Message with invalid phone number. |
| 36 | Played with receipt | Message reception confirmed. |
| 37 | Connection timed out | No connection within defined time range. |
| 38 | In progress | Receiver is being called. |
| 39 | Next try in 15 min | Receiver could not be reached, retry pending. |

– Table: Status Codes Voice

3.4.3.3. Status codes fax messages

| Fax | | |
|--------------------|-------------------------------|---|
| Status Code | Status Text | Description |
| 41 | Ok | Fax message successfully transmitted. |
| 43 | Waiting | Transmission waiting. |
| 44 | New job | Request newly issued. |
| 101 | Applied | Job group successfully assigned to gateway, |
| 201 | Started converting | Started converting. |
| 202 | Finished converting | Single file converted. |
| 301 | Started merging | Merging documents of fax message. |
| 401 | Processing | Fax message is being processed. |
| 402 | Finished processing job | Processing single job finished. |
| 501 | Finished processing job group | Processing job group finished. |
| 5513 | Exists in the blacklist | Callback, address or parts of the content are blacklisted |
| 6000 | Unknown error | Unknown error. |
| 6002 | Conversion error | Error converting the documents. |
| 6004 | No response | No response. |
| 6005 | Empty job | Error: Empty order. Cannot create JobID. |

| Fax | | |
|--------------------|---------------------------|--|
| Status Code | Status Text | Description |
| 6006 | Line busy | Line busy. |
| 6007 | Rejected | Call rejected. |
| 6009 | Unknown number | Unknown call number. |
| 6010 | Invalid number | Invalid call number. |
| 6011 | Number changed | Number changed. |
| 6013 | No fax device | The receiver is not a fax device. |
| 6014 | Disconnected by sender | Connection dropped by sender. |
| 6015 | Disconnected by receiver | Connection dropped by receiver. |
| 6016 | File format not supported | File format not supported. |
| 6017 | File access failed | No access to file. |
| 6900 | Ok (not confirmed) | Transmission ok, connection dropped by receiver. |

– Table: Status Codes Fax

3.5. Message Buffer

For the commands `GetNextMessage` and `DeleteMessage`, the buffer must be activated on the website. Only then the incoming messages will be written into the memory. Messages already received cannot be incorporated retrospectively. The memory can be set up under *Account Settings > Virtual mobile number > 3. Forward SMS's received (optional)* as a forwarding option.

3.5.1. Query for new messages

This command reads the oldest message received on a leased SMS number. In addition, a unique identification number is also delivered, which can be re-used for deletion from the buffer.

As an option, the leased mobile phone number can be indicated. In this case, the oldest message sent to this number is returned. If this option is missing, the oldest message for all numbers leased by this account is returned.

- **Function**

`Api/Cache/GetNextMessage` Function for reading the next message from the buffer

- **Compulsory Parameters**

`UserName` Account username or self-created API credentials (*Interfaces > Overview*)

`Password` User password.

- **Optional Parameters**

`MobileNumber` Leased SMS number in international format (e.g.: 0041763332020) or in national format (e.g.: 0763332020).

- **Examples**

```
/Api/Cache/GetNextMessage?Username=xxx&Password=yyy
```

```
/Api/Cache/GetNextMessage?Username=xxx&Password=yyy&MobileNumber=+41763331111
```

- **Reply for GetNextMessage**

The reply structures here are no longer based on the HTTP status code, rather they have an XML structure as their basis.

Structure (without content)

```

<IncomingSMS>
  <Result>
    <ResultCode></ResultCode>          (as per paragraph Error Codes)
    <ResultText></ResultText>         (Description of the error code)
  </Result>
  <MessageID></MessageID>            (→ Use to delete the message)
  <SMS>
    <Message></Message>
    <AddressTo></AddressTo>          (internat. format)
    <AddressFrom></AddressFrom>      (internat. format)
    <TimeStamp></TimeStamp>         (Format: dd.mm.yyyy hh:mm:ss)
  </SMS>
</IncomingSMS>

```

Note: If there is no message in the message buffer, the ResultCode 11204 is returned (no new incoming message present") and the SMS and MessageID nodes are NOT shown.

3.5.2. Delete messages from the buffer

This command deletes the message from the buffer, but not from the account. The message is still visible on the website at “Messages received”.

- **Function**

Api/Cache/DeleteMessage Function for deleting a message from the buffer

- **Compulsory Parameters**

UserName Account username or self-created API credentials (*Interfaces > Overview*)

Password User password.

MessageID ID of a message received from the function *GetNextMessage* ([Status query for a particular](#))

- **Examples**

/Api/Cache/DeleteMessage?UserName=xxx&Password=yyy&MessageID=123abc

- **Reply for DeleteMessage**

This reply structure describes whether a message was successfully deleted from the buffer. It looks like this:

Structure (without content)

```
<DeleteMsg>
  <Result>
    <ResultCode></ResultCode>
    <ResultText></ResultText>
  </Result>
</DeleteMsg>
```

3.6. Receiving replies to an SMS

- **Function**

Answer Reply message function.

- **Compulsory Parameters**

AddressTo Recipient number

AddressFrom Sender number

Message Content of the reply

Note: Windows-1252 encoding is used for the message.

TimeStamp Time of receipt (dd.mm.yyyy hh:mm:ss).

- **Optional Parameters**

JobID Contains the unique identification of the message if it was included in the sending job.

- **Read confirmation**

You should respond with HTTP status code 200 OK to confirm that you successfully received the reply.

- **Example**

<http://www.example.com/answers?Function=Answer&AddressTo=0041766012999&AddressFrom=0041791234567&Message=My+Reply&TimeStamp=11%2E12%2E2015+10%3A37%3A00>

3.7. Notification message

- **Function**

Notification Notification message function.

- **Required Parameters**

ResultCode Status code of the message (as shown in the notification table).

ResultText Status in plain text (as shown in the notification table).

Number Contains the recipient number for this notification.

TimeStamp Contains the receipt time for the sent message (dd.mm.yyyy hh:mm:ss).

- **Optional Parameters**

JobID Contains the unique identification of the message if it was supplied when sending the job.

- **Read confirmation**

You should respond with HTTP status code 200 OK to confirm that you successfully have received the notification message.

- **Example**

<http://www.example.com/notifcations?Function=Notification&ResultCode=0&ResultText=Message+has+been+delivered&TimeStamp=1Z%2E07%2E2014+12%3A55%3A15&Number=0041787557539&JobID:FB551638-A320-476F-A47D-F2EE68>

3.7.1. Notification result codes

| ResultCode | ResultText | Description |
|------------|--------------------------------|---|
| 0 | Message has been delivered | Receipt of the forwarded message was confirmed by the recipient. |
| 1 | Message has been buffered | Receipt of the forwarded message was confirmed by the recipient. |
| 2 | Message has not been delivered | Receipt of the forwarded message could not be confirmed. |
| 3 | Error Code / Error Message | Defines the error on transfer to the corresponding central point. |
| 4 | Transmission OK | The send job was forwarded to the corresponding central point. |

– Table: Notification Result Codes

3.8. Forwarding received SMS messages to HTTP

In the eCall web portal, under the settings where SMS numbers can be rented, there is the option to have incoming SMS forwarded to an HTTP(S) address. The URL eCall should forward received messages to must be configured there as well (e.g.: <https://mycompany.com/incoming/sms>).

Incoming messages to a virtual number may be forwarded via GET request or via POST request. For GET calls, eCall supplements this URL with various parameters. POST requests use the JSON format. Both forwarding structures are described in this chapter.

3.8.1. Forwarding via GET requests

eCall supplements the configured URL with various parameters. The following parameters are included:

| | |
|--------------------|--|
| Function | Always contains IncomingSMS . |
| AddressTo | The SMS was received on this eCall reception number. The forwarding was stored for this number. |
| AddressFrom | Sender number. The SMS was sent from this number. |
| Message | Content of the received SMS. |
| TimeStamp | Time of receipt in the eCall system. The format corresponds to the following pattern: "DD.MM.YYYY HH:MM:SS". |

Since all parameters are included in the URL, special characters are coded accordingly. For example, a space is coded with "%20", a colon with "%3A".

- **Example:**

```
...?AddressFrom=0041799999999&AddressTo=0041766666666&Function=IncomingSMS&Message=Yes%2C%20noon%20is%20ok.&TimeStamp=01.01.2020%2012%3A00%3A00
```

would mean

```
Function      IncomingSMS
AddressTo     0041766666666
AddressFrom   0041799999999
Message       Yes, noon is ok.
TimeStamp     01.01.2020 12:00:00
```

3.8.2. Forwarding via POST requests

When an incoming message is forwarded using POST, the request body contains a JSON string in the following format:

```
{
```

```

    "messageId":"string",
    "channel":"Sms",
    "to":"string",
    "from":"string",
    "content":{
      "text":"string",
      "type":"Text"
    },
    "timestamp":"2024-03-13T11:44:40Z"
  }
}

```

The meaning of the parameters is as follows:

| | |
|------------------|---|
| messageId | GUID for unique identification of the incoming message. |
| channel | Always «Sms». |
| to | The SMS was received on this eCall reception number. The forwarding was stored for this number. |
| from | Sender number. The SMS was sent from this number. |
| content | <p>type Always «Text».</p> <p>text Content of the received SMS.</p> |
| timestamp | Time of receipt in the eCall system as UTC. The format complies with ISO 8601. |

- **Example:**

```

{
  "messageId":"6815c501-2fe1-ee11-812a-00155dfb061c",
  "channel":"Sms",
  "to":"00417666666666",
  "from":"00417777777777",
  "content":{
    "text":"Hello eCall!",
    "type":"Text"
  },
  "timestamp":"2024-03-13T11:44:40Z"
}

```

4. Reply codes

There are two types of reply codes. For message dispatch, the status is returned directly as an HTTP status code and additional ResultCodes, while for operations with the buffer, only the eCall ResultCodes will be returned.

4.1. HTTP status codes

HTTP Status codes are used when sending jobs, (Chapter 3.1 to 3.3).

Status codes 4xx generally indicate a client error and 5xx indicate a server error.

The following codes may be returned:

| Status code | Status text | Description |
|-------------|---------------------|---|
| 200 | OK | Job received successfully |
| 500 | InternalServerError | General error during processing |
| 400 | BadRequest | This code is returned for various reasons. Please refer to the eCall ResultCodes table below. |
| 401 | Unauthorized | The credentials used are unknown or missing (Username/Password) |
| 403 | Forbidden | Insufficient points available on the account |

– Table: HTTP Status Codes

For querying messages, status code 200 (OK) is returned. The operation executed is described in the body text using an XML (For structure, see [Message Buffer](#)). The ResultCodes from are used for this.

4.2. eCall ResultCodes

| Code | ResponseText | Description |
|-------|--------------------------|---|
| 0 | OK | The message was sent |
| 11000 | SyntaxError | Syntax error |
| 11001 | PermissionDenied | Access denied |
| 11100 | AdrAdCInvalid | Invalid or incorrect recipient address |
| 11101 | AdrNAdInvalid | Invalid or incorrect confirmation address |
| 11102 | AdrAdCMissing | Missing recipient address |
| 11103 | AdrNAdMissing | Missing notification address |
| 11104 | AdrAdCTooMany | Too many recipient addresses transmitted |
| 11105 | AdrAdCNotAllowed | Recipient address not within defined area |
| 11200 | MsgNoCharacters | No message existent |
| 11201 | MsgInvalidCharacters | Invalid characters in the message |
| 11202 | MsgSendTimeInvalid | Invalid sending time given |
| 11203 | MsgSendTimeNotAllowed | Sending time is not within time window |
| 11204 | MsgTooLong | SMS/Pager only: Message is too long. In this case you have to check the setting „maximum number of pages for long messages“ in eCall. |
| 11300 | AccNoUser | Unknown User |
| 11301 | AccNoUnits | Not enough credits in account |
| 11303 | AccNoFreeUnits | Not enough free credits in account |
| 11400 | CallSystemInvalid | Invalid call system |
| 11401 | CallSystemInvalidForUser | Invalid call system for this user |
| 11402 | CallSystemNotSupported | Call system is not supported |
| 11403 | CallSystemConfused | Call system is temporarily out of order |
| 11500 | IDMissing | ID is missing |
| 11501 | IDInvalid | ID could not be found |
| 11502 | IDDoesNotExist | ID does not exist in the system |
| 11600 | StateAlreadyTransmitted | Message was already transmitted |
| 11700 | DataReadError | Data contents could not be read |
| 11800 | CBMsgError | Callback contains a forbidden number or text |
| 11810 | AnswerInvalid | Answer parameter contains an invalid address |
| 11904 | Too many attachments | Too many attachemts uploaded (max. 10 allowed) |
| 11905 | Attachment(s) too large | At least 1 attachment is too large. Max. size for attachments is 5 MB |

| | | |
|-------|--|--|
| 11906 | File type not supported | File type is not supported as attachment. The following types are supported by eCall: bmp, pdf, doc, rtf, ppt, dok, snp, gif, tif, tiff, html, txt, jpg, wir, jpeg, xls, zip |
| 11907 | Error while processing the attachments | There was an unexpected internal error while processing the attachments |
| 11908 | Error while saving the attachments | There was an unexpected internal error while saving the attachments. |
| 11910 | Job not found in Log | Job could not be found in the log |
| 11911 | JobID required | In a job status request you have to submit a JobID |
| 11912 | Job is scheduled | Job is not executed yet but scheduled |
| 11913 | Job in progress | Job is currently in progress |
| 11914 | Attachment name exceeds char limit | The name of the attachment is too long |
| 11999 | Unknown error | Unknown error |

– Table: Result Codes

5. Connections

5.1. *Access addresses for sending jobs*

The URL access for eCall is as follows:

<https://url.ecall.ch/>

The service is only available over an encrypted connection using TLS 1.2.

5.2. *IP addresses for replies or notifications (eCall to client)*

Source IP Address: 193.93.208.200
193.93.208.149
193.93.208.153

Source port: undefined (open; 0 to 65535)

6. Header information

6.1. *Client Header ID*

To help you matching the responses to the sent requests more comfortable, it is possible to set the following header information (max. 100 characters):

x-f24-client-request-id

If this header parameter is set, it will be included into the response header, too.

6.2. *Server Header ID*

Each request made against the eCall REST API returns a response header named **x-f24-request-id**. This header contains an unique value that identifies the request.

If a request is consistently failing, and you have verified that the request is properly formulated, you can use this value in support cases. In your report, include the following information:

- The value of **x-f24-request-id**, as received in the response.
- The approximate time that the request was made.
- Your eCall account name and/or interface user you tried to send with.
- The type of operation that the request attempted (e.g. sending a message).

7. Access test

You can use your web browser for a simple access test. Please note the [requirements](#) for access use. In addition, you must know if your username and password are set up for identification.

1. Open your browser.
2. Type the URL below into the address bar of your browser.

<https://url.ecall.ch/api/sms?username=maxmuster&password=s3cr3t&address=0041791234567&message=Dies%20ist%20ein%20Test>

<https://url.ecall.ch/Api/Sms?Address=%2B41791112233&Message=Test&Username=foo&Password=bar>

Please note that you must replace the values for “Username”, “Password” and “Address” with your own details.

3. You will receive a reply as described in Reply codes

8. Table directory

| | |
|--|----|
| – Table: Status Codes SMS / Pager | 16 |
| – Table: Status Codes Voice | 17 |
| – Table: Status Codes Fax | 19 |
| – Table: Notification Result Codes | 25 |
| – Table: HTTP Status Codes | 28 |
| – Table: Result Codes | 30 |