

# O<sub>2</sub> SMS Connector

Technical description of Web services interface



## Content

1	Introduction.....	3
1.1	Purpose and Scope .....	3
1.2	Definitions, Acronyms and Abbreviations.....	3
2	System Context and usage Scenarios.....	4
2.1	The O2 SMS Connector service .....	5
3	Interactional Behaviour of the System .....	5
3.1	General Behaviour / Error responses.....	5
3.2	Sending Messages (AO-MT) .....	6
3.3	Receiving MO-AT Messages .....	7
4	Applicable Standards.....	8
5	Functional Description of the Interface .....	8
5.1	HTTP Get / Post Interface Basics .....	8
5.2	BA Input Parameters.....	9
5.2.1	Parameter Mappings.....	13
5.3	Output Format.....	14
5.4	MP Output Parameters .....	14
5.4.1	Parameter Mappings.....	16
5.5	Send .....	17
5.6	Receive.....	17
5.7	Confirm .....	17
5.8	Examples of Requests and Responses .....	18
5.8.1	send .....	18
5.8.2	POST .....	18
5.8.3	receive .....	19
5.8.4	confirm .....	21
5.9	Response Types and Codes.....	22
6	Security.....	23
7	Cookies.....	23
8	Registering a business application .....	23
9	Appendix C: Deployment Specific Parameters .....	32
10	Appendix D: Solution constraints .....	32



## 1 Introduction

### 1.1 Purpose and Scope

This document describes the HTTP GET/POST protocol for the O2 messaging platform and is intended for use by business partners to develop business applications capable of sending text SMSs and receiving both text and binary SMSs.

For the HTTP GET/POST protocol O2 Messaging Platform offers the SMS Connector service, which includes the secure sending of text SMSs and the receiving of text SMSs and binary SMSs including automated provisioning and billing. Because of the limited capabilities of the protocol only a subset of the services functionality (e.g. no sending of binary messages) is supported. If you wish to use the full functionality of the messaging platform, O2 offers a Web Services interface; please refer to the respective O2 documentation.

### 1.2 Definitions, Acronyms and Abbreviations

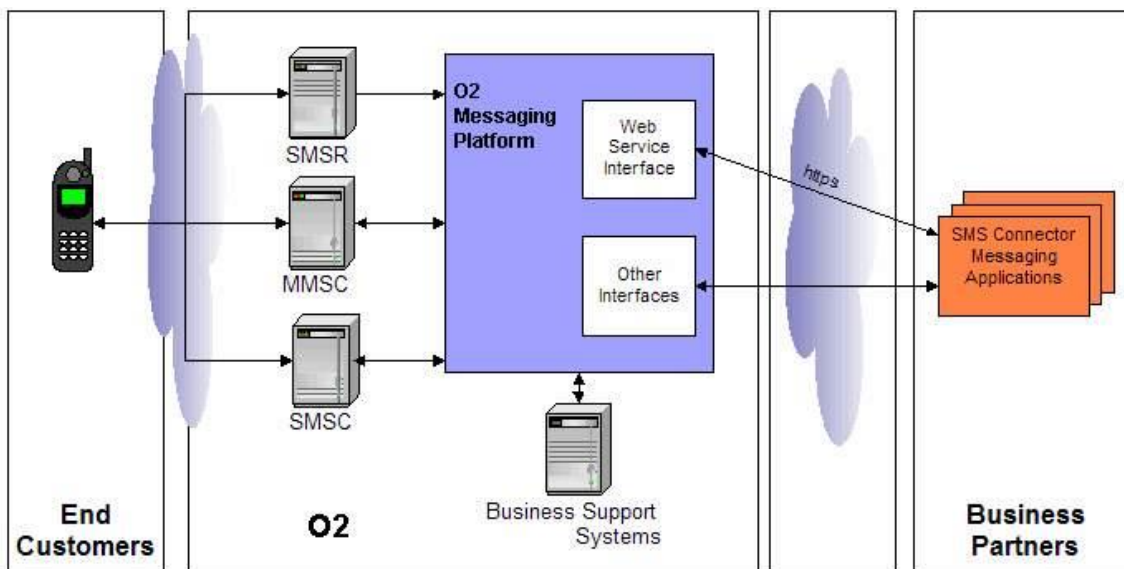
Abbreviation	Definition
MP	Messaging Platform - A logical system made up of the Web Services Gateway, HTTP Server, and other internal O2 systems.
BA Profile	This is data for a business application that comes from the provisioning system. This includes everything the Messaging Platform things such as capacity, certificates, etc.
CDR	Call Detail Record
BA	Business Application
MSISDN	The mobile user's telephone number.
MO	Mobile-Originated - is a term used to determine the identification of the source direction or originator of a message.
MT	Mobile-Terminated - is a term used to determine the identification of the target direction or destination of a message.
AO	Application-Originated - is a term used to determine the identification of the source direction or originator of a message.
AT	Application-Terminated - is a term used to determine the identification of the target direction or destination of a message.
SMSC	SMS Center – The O2 network element that the MP sends SMS messages to and receives SMS messages from. Although ET has multiple SMSCs, from the perspective of this document, there is no difference and hence they are all grouped into this term.
SOAP	Simple Object Access Protocol – An XML standard that is used to define Web Services.
Web Services	An XML standard based on SOAP that defines the way applications can speak with one another.
CA	Certificate Authority
API	Application Program Interface



UDDI	Universal Description, Discovery and Integration registry – used to store Web Service definitions that can be used by applications to find information about how to connect to a Web Service.
SMS Connector	This is the product that business partners will use to send and receive SMS messages.
SPK	Service Packages.
RPC	Remote Procedure Call
WSGW	Web Services Gateway
WSDL	Web Services Definition Language.
WSIF	Web Services Invocation Framework
XML	Extensible Markup Language
XSD	XML Schema Definition
SMRS	SMS Redirector - O2 network feature enabling BA to receive MO SMS from other operator networks

## 2 System Context and usage Scenarios

The O2 Messaging Platform provides business applications that are operated by business partners with a highly scalable and reliable two-way messaging channel.



In the diagram above the main entities are:

- O2: offers the messaging services and is responsible for Messaging Delivery, Billing and Customer Care. O2 exposes this service via the HTTP GET/POST protocol as described in this document.
- Business Partners: organizations that send and receive short messages by using the O2 Messaging Platform. The business partners implement business applications for the O2 SMS Connector service.
- End Customer: an individual who receives/sends short messages from/to Business Application of a O2 Business Partner. Receiving/sending SMS from/to End Customer using fix line is not supported.

O2 offers different interface protocols (HTTP GET/POST, Web Services) to partner business applications. This document refers to the HTTP GET/POST interface only.

The service is designed in such a way, that it is possible for a business application to use all underlying interface types, e.g. HTTP GET/POST for sending and Web Services for receiving messages. The restriction is that all processing steps related to the receiving of one message (receive and then confirm) should be done by using the same interface.

Despite this restriction, all other combinations are valid as long as synchronous communication (i.e. receive and confirm) is done through the same interface, for example the BA can send a message via the Web Services interface and receive the corresponding delivery report using the HTTP GET/POST interface. Please see chapter 3 for more details.

The HTTP GET/POST interface acts as a proxy between the internal O2 messaging platform and the business application. It provides a secure method of sending and receiving messages by using HTTPS and client side authentication (see chapter 6 Security). The interface allows platform-independent access to the messaging services.

## 2.1 The O2 SMS Connector service

This product is for use by O2 Business Partners (BP) wishing to send and receive SMS messages to mobile subscribers. The platform supports text and binary SMS messages, but when using this interface BAs may receive both text and binary SMSs but they may only send text SMSs. The goal of this product is to deliver a platform which will connect the BP's business application to O2's messaging platform through a set of standardized interfaces including Web Services (see relevant specification) and HTTP GET/POST. Although the service is designed to allow business applications to communicate with the platform using either of the two interfaces (HTTP and Web Services), business applications must perform the receive-confirm sequence (See section 3.3) using the same interface. In addition, O2 is responsible for automated provisioning and billing.

The following chapters describe the interface's functionality and its behavior.

## 3 Interactional Behaviour of the System

The platform offers two primary services to Business Applications:

- send – simple or multi-part text content to a target mobile device (AO message)
- receive – receive text or binary SMSs from mobile devices (MO messages).

### 3.1 General Behaviour / Error responses

The MP interface uses a generic messaging protocol offering a means of transporting messages and responses asynchronously. The MP returns the status in a synchronous or asynchronous fashion. Synchronous means the response stream arriving directly as result of the called action (e.g. send). Asynchronous means that the MP sends a message to the BA's message queue after the originating call has been finished (an example of this would be a delivery report). Asynchronous responses are received like for example text SMSs. The different error types and codes which may happen are explained in chapter 5.9.

The following type of messages can be received from the MP asynchronously:

- MO messages (text SMS or binary SMS)
- Asynchronous responses (typically errors related to AO )
  - Special kind of asynchronous responses: Delivery reports in case these were requested by BA for AO messages

Each message is uniquely identified by a message ID (unique related to the BA ID). For AO messages this ID may be provided by the Business Application and then the BA must ensure that the ID is unique for it. If the ID is not provided by the BA, it will be generated by the MP. For MO messages and asynchronous responses, the MP is responsible for generating the unique IDs.

In case messages are not retrieved by the BA these will be deleted after a certain time period preset by O2 (deployment specific parameter `messageReceptionTimeout`; see chapter 11 for more detail).



The following is common for all communication processes originated from the business application side: The MP performs the authentication of the BA (validity of the certificate and valid reference to the BA ID). This is not explicitly shown in all flow diagrams for better readability. Please refer to the security chapter 6 for more details.

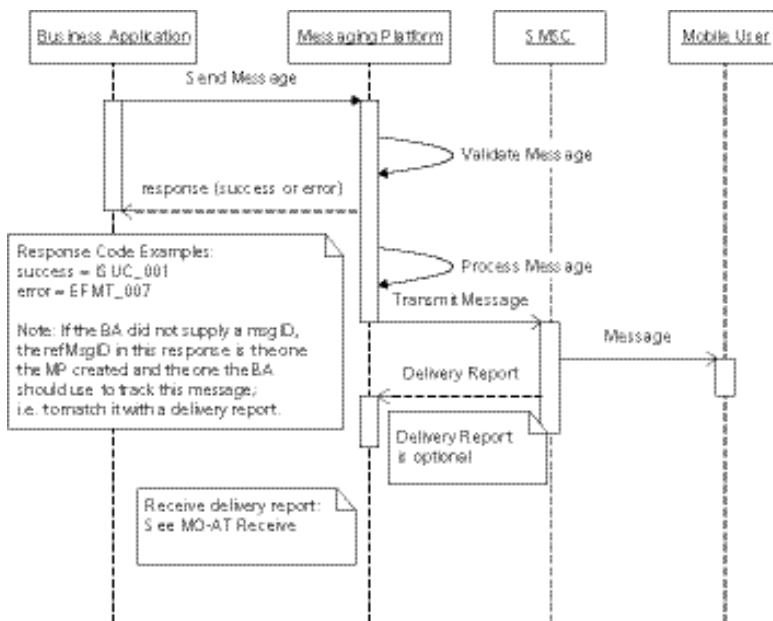
### 3.2 Sending Messages (AO-MT)

The BA sends a message to the MP using the send action. The MP performs only the basic checks synchronously and returns an HTTP OK response for success (if message was accepted) or an HTTP error response (the message is not accepted and will not be processed). Additional responses may be produced during the processing of the message and these are sent to the BA asynchronously. See chapter 9 for more detailed information about response codes.

The BA may request a delivery report. When the message is successfully delivered the MP sends a delivery report message to the BA, otherwise a non-delivery report is sent. The business application may incur additional charges for a delivery report. Delivery reports and asynchronous responses are retrieved by the BA using the same method as for a regular MO-AT message. If the BA requests a delivery report for a multipart SMS, the delivery report will be successful only if all parts arrive at the mobile. If one or more parts do not arrive, the delivery report will return false.

There is a maximum AO message throughput for each BA and the Messaging Platform has a throttling mechanism to ensure this limit is not exceeded. If this limit is exceeded calls to the send action will slow down.

The following diagram shows the typical message flow:



The MP allows BA to process messages in multiple threads. However there is a limit to the number of threads each BA is allowed for each method. If the limit is exceeded the action is rejected and a synchronous response with an error code, such as EAPP\_053, is returned to the BA. The actual thread limits are deployment specific parameters: sendThreadLimit, receiveThreadLimit and confirmThreadLimit, and maxCommunicationThreads.

### 3.3 Receiving MO-AT Messages

For receiving messages the BA connects to the MP by invoking the receive action. If a message is available the MP responds by sending the first message from the delivery queue. If the platform has no messages in the BA queue, the call will be blocked for a timeout period defined by O2 (deployment specific parameter `receiveBlockPeriod`; see chapter 11 for more detail) or until a message is available. If a new message arrives in the BA queue during the block period, the message will be returned to the BA immediately. If no message is available and the timeout is exceeded, the BA will receive an empty HTTP response. See chapters 5.8, 5.8 (POST) for examples. This response does not require confirmation.

The BA confirms the receipt of a message by invoking the confirm action of the messaging platform. After confirmation the messaging platform is able to delete the message from its delivery queue. If the BA could not receive a message because the MP queue is already empty or for some other reason (e.g. when an error occurs), it does not need to confirm the MP response.

If one or more messages are not retrieved by the BA these will be deleted after a certain time period preset by O2 (deployment specific parameter `messageReceptionTimeout`; see chapter 11 for more detail)

If a message was received but not confirmed within a certain time (deployment specific parameter `confirmationTimeout`; see chapter 11 for more detail) it is put back on the front of the queue available for polling. Please note that the overall message expiration timeout (deployment specific parameter `messageReceptionTimeout`) is completely independent of the receive-confirm process. This means that the number of times a BA receives a message but does not confirm it, has no impact on when the message will expire and be deleted from the MP.

If there are several messages in the delivery queue, the business application repeats the receive-confirm sequence until the messaging platform returns that there are no more messages. Messages are delivered according to the First In First Out (FIFO) principle: all messages are processed by the messaging platform in the order of their arrival. The order of delivery to the BA, however, is not guaranteed as a BA may use parallel threads to receive messages from the MP.

It is also possible for the BA to receive messages using a number of parallel threads. There is a limitation to the number of maximum allowed communication threads for one BA that is configured in the MP. If there are more communication threads opened than allowed, the MP will return a synchronous response notifying the BA. The BA also has to be aware that only a limited number of messages (deployment specific parameter `maxUnconfirmedMessages`; see chapter 11 for more detail) can be received without confirmation. The first thread that tries to receive a message exceeding the limit will receive an error message until at least one of the received messages is confirmed or the confirmation timeout is exceeded.

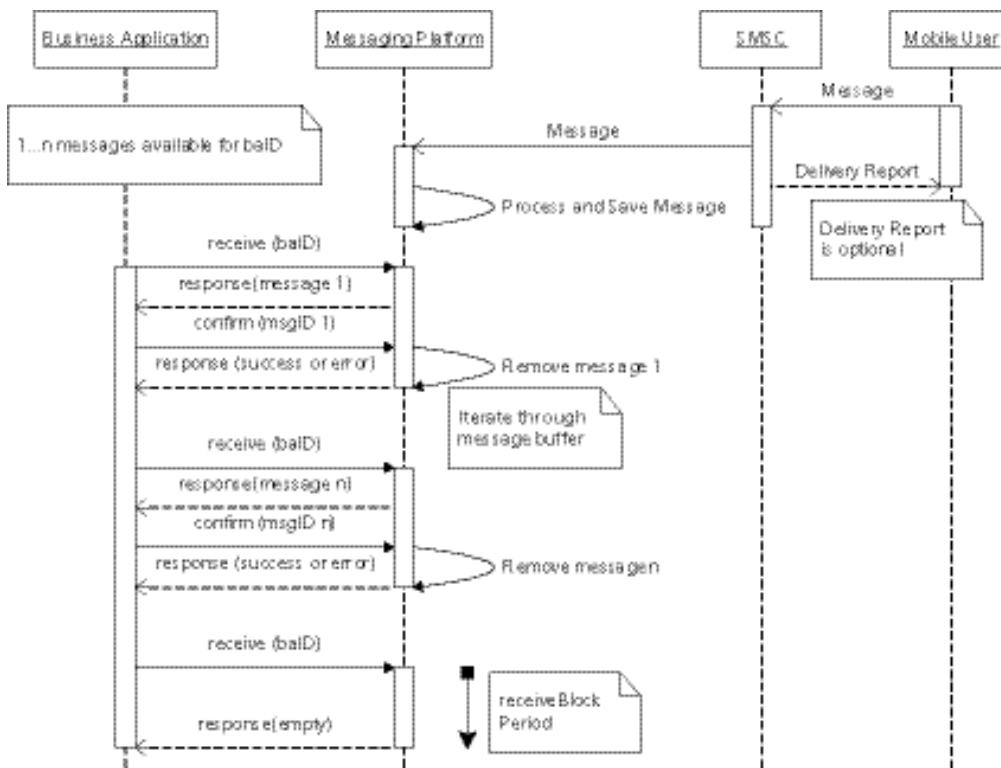
To make parallel threads possible for the receive-confirm mechanism the MP and the BA have to be able to identify specific messages. Therefore the reference message ID, reference BA ID and the BA ID parameters need to be used to confirm a message. When calling the receive method, the BA passes its BA ID to the MP. In the MP's response, which contains the message, there is also given back the message ID in the HTTP response. The BA must use this message ID when calling the confirm action to identify the message in the MP queue. Please see chapter 5 for more details.

BA should not confirm synchronous responses returned during receive action. This typically indicates technical errors related to the incorrect behaviour of the BA, such as too many messages received and awaiting confirmation (response code `EAPP_037`).

Synchronous responses can easily be distinguished from the asynchronous ones, as the parameter `refMsgID` will be empty for synchronous responses.

The following diagram depicts the typical message flow:





## 4 Applicable Standards

The interface builds on the HTTP 1.1 standard (W3 Network Working Group RFC2616). For security reasons the SSL protocol with client-side authentication (x.509v3 client certificates) is used.

## 5 Functional Description of the Interface

This chapter describes the O2 messaging HTTP interface actions and data. The MP HTTP GET/POST interface offers three basic actions:

- send for sending of AO text SMSs
- receive for retrieving text SMSs and binary SMSs from the MP
- confirm for confirmation, that a certain message has been received successfully

The following chapters explain the input and output parameters, the methods' outputs and the methods themselves.

### 5.1 HTTP Get / Post Interface Basics

The functionality is used by specifying the HTTP GET or POST methods as specified in the HTTP 1.1 standard (W3 Network Working Group RFC2616).



The interface supports the HTTP POST and GET methods. The BA should be aware that all parameters have to be sent URL-encoded in the request when calling the message platform. Please note that the GET method is limited to 1024 bytes in total length.

We recommend using the POST method. For the POST method the HTTP header content-type must be set to „application/x-www-form-urlencoded“. For the specification on URL-encoding please refer to RFC-1738.

Access to SMS Connector is provided through the following URL:

<https://smsconnector.cz.o2.com/smsconnector/getpost/GP>

For limited time period older URLs will be also supported, but business partners are advised and kindly asked to use new O2 URL.

Older URL is:

<https://smsconnector.eurotel.cz/smsconnector/getpost/GP>

## 5.2 BA Input Parameters

The following table gives an overview of the input parameters for the different actions. The parameters will be content of the HTTP request either using the Get or the Post method.

For examples see section 5.8 below.

All parameter names are case insensitive. Parameter values are case sensitive except values for parameters 'action', 'deliveryReport', and 'multipart'. Values for named parameters are case insensitive. Please note that if a parameter is missing it is considered as null by the MP. If the parameter is included but the value is empty, such as "abc=", the value of the parameter will be checked and used by the MP. In this case, it is possible for the MP to reject the action because of an invalid (i.e.missing) parameter value. So, if the BA wants to use a default, it should not include the parameter.

Legend:

X: mandatory field

O: optional field

-: field not supported for that action

Parametr	send	receive	confirm	length	description	default value
action	X	X	X		Action specification send - send TextSMS receive - receive message from message queue confirm - confirm and delete specified message	--
baID	X	X	X	6 or 7	Business Application ID, 7 digits identifier of the Business Application from O2's 2-Way SMS numbering range, MSISDN assigned to the application starting with 199. E.g. 1991123. May be 6 digits in case of application migrated from the BMG service (starts with 99).	--
toNumber	X	-	-	max 16	Destination number of the SMS. The MSISDN in	--



international format with a leading “+”, country code, and a national number. An example for a Czech number is +420602001234. For Czech numbers, the MP will check that the number is exactly 12 digits long, including country code, so +420123456789.

text	X	-	-	max 900	Text of message with multipart message support; the max length is unlimited (in case Multipart=TRUE or 160 characters (in case Multipart=FALSE). The maximum length for multipart text SMS messages is 900 characters. BAs are responsible for compatibility of AO multipart text SMS messages with mobile subscribers' handsets. In AO direction, all characters must have their counterparts in 7-bit default GSM encoding (defined in GSM 03.38), which also means that characters with Czech diacritics are not supported.	--
Parametr	send	receive	confirm	length	description	default value
msgID	O	-	-	max 128	Unique message ID for all messages of the BA. For AO messages it may be provided by the BA; in case the field is not filled, a unique ID is generated by the MP. A recommendation is to include timestamp in the message ID. To guarantee proper processing of the message, message ID should not contain semicolon (“;”) character.	If null msgID is generated by the MP
deliveryReport	O	-	-	5	Request for a delivery report: FALSE - No delivery report, TRUE - Delivery report requested	FALSE

For setting correct setting of delivery reports see also description of field validityPeriod and also Appendix Solution Constraints.

intruder	O	-	-	5	TRUE - message is displayed on mobile phone FALSE - message will be sent as standard text SMS	FALSE display
multipart	O	-	-	5	Indicates that the maximum text/data size can be exceeded, and the platform will divide the message into necessary number of parts. The division algorithm simply splits the messages every 154 characters (if they are more than 160 characters long). For phones that support concatenation, the message will be reassembled into one. On phones that do not support this feature, it is possible for words to be split in the middle. O2 will charge for each part of the message. Specification of multipart message (TEXT length >160): FALSE - standard SMS with TEXT length max. 160 TRUE - multipart SMS with TEXT length max. 900	FALSE
Parametr	send	receive	confirm	length	description	default value
suffix	O	-	-	max 9	BA Suffix number with up to 9 digits; is appended to the BAID as originating number and displayed at the destination mobile. Parameter 'suffix' must not be present if parameter 'fromNumber' is present.	--
validityPeri	O	-	-		Validity period of the message measured in seconds measured from the time MP received the message from BA. MP calculates absolute time when message should expire by adding validity period to the system time of message receipt. This absolute	0

expiration time is then managed by SMSC. The value could be modified (rounded or shortened) by the MP or the SMSC while processing. If message cannot be delivered to the mobile user by the given expiration time, SMSC will discard the message. If validity period is set to "-1", the message is sent immediately and only once. If value is '0' (the default when no value is specified) the validity period is managed by MP. For correct setting of validityPeriod value see also Appendix Solution Constraints.

priority	O	-	-	1	Message priority: 1 - VIP message 2 - high 3 - normal	3
refMsgID	-	-	X	max 128	The message ID (msgID) of the message that is being confirmed.	--
refBaID	-	-	X	6 or 7	Business Application ID, 7 digits identifier of the Business Application from O2's 2-Way SMS numbering range, MSISDN assigned to the application starting with 199. E.g. 1991123. May be 6 digits in case of application migrated from the BMG service (starts with 99). This should be the same value as the baID of the message that is being confirmed.	--

Parametr	send	receive	confirm	length	description	default value
fromNumb	O	-	X	max 16	Originating number of the SMS. It can be either the BA ID plus the optional suffix (up to 9 digits long) or MSISDN assigned to the BA by O2 in international format with a leading "+", country code, and a national number. An example for a Czech number is +420720001001.	

If parameter 'fromNumber' is present the AO message



is sent from 'fromNumber' value. If the parameter is not present, the message sender is created from values in 'baID' + 'suffix' fields. Parameter 'fromNumber' must not be present if parameter 'suffix' is present.

### 5.2.1 Parameter Mappings

This section describes which parameters are required, optional or not supported for each type of object. Please note that sending binary SMSs is not supported.

Legend:

X: mandatory field

O: optional field

-: field not supported for that action

Parameter	TextSms
baID	X
toNumber	X
text	X
msgID	O
deliveryReport	O
intruder	O
multipart	O
suffix	O
validityPeriod	O
priority	O
refMsgID	-
refBaID	-
fromNumber	O

### 5.3 Output Format

The synchronous result of the operation will be returned in the body of the HTTP response. It will contain the message data as specified in the following sections or the respective error information in the fields:

responseType, responseCode, and responseDescription.

Synchronous errors will be returned with an HTTP status code of 400 (bad request). Please note that this response can also be returned from the receive and confirm actions (e.g. too many concurrent requests).

The format will be a parseable String consisting of multiple lines. Each line contains one output parameter including value, which follows the following convention:

```
OUTPUT_PARAMETER_NAME=VALUE{LineBreak}
```

No extra space character is added before or after “=” character. The number of lines in a response depends on the operation and its result.

Because the value of the „text“ parameter may contain line breaks, it may spread across more than one line. Therefore it will always be the last parameter in the message.  
Please see chapter 5.8 for examples..

### 5.4 MP Output Parameters

Please note that all responses will be in UTF-8 encoding.

The MP will return the following parameters depending upon the type of operation:

Legend:

X: will be returned

O: may be returned depending on input values and error condition

-: return value not supported for that action

Parametr	send	receive	confirm	description
selector	X	X	X	Type of received message: TextSms - text SMS message received BinarySms - binary SMS message received
msgID	X	X	X	Response – status report (e.g. delivery report or error report) This field contains the unique message ID generated by the MP (for AO may also be generated by the BA). In case of authorization errors in the send action this field is empty.
refMsgID	X	O	X	Message ID of the original message to which the received message (e.g. delivery report) refers to. Parameter will be empty for synchronous responses.

Parametr	send	receive	confirm	description
fromNumber	-	O	-	<p>Originating number of the SMS. The formats differ for MSISDNs and for applications as described below:</p> <ul style="list-style-type: none"> <li>• The MSISDN in international format with a leading "+", country code, and a national number. An example for a Czech number is +420602001234.</li> <li>• The application number is either 5-16 digits, starting with the baID (e.g. 1991123) or the MSISDN assigned to the application in international format with a leading "+", country code, and a national number (e.g. +420720001001) Is empty for type report.</li> </ul>
toNumber	-	X (empty if serious error)	-	<p>Destination number of the SMS. The formats differ for MSISDNs and for applications as described below:</p> <ul style="list-style-type: none"> <li>• The MSISDN in international format with a leading "+", country code, and a national number. An example for a Czech number is +420602001234.</li> <li>• The application number is either 5-16 digits, starting with the baID (e.g. 1991123) or the MSISDN assigned to the application in international format with a leading "+", country code, and a national number (e.g.+420720001001)</li> </ul>
timestamp	X	X	X	<p>Time when message was sent to the platform (local time) or when the response was generated. Format e.g. 2003-06-15T20:45:11</p>
text	-	O	-	<p>Body of the text message with a maximum length of 160 characters.</p>
responseType	X	O	X	<p>Only for the type Response: it tells the BA what kind of response this is.</p>
responseCode	X	O	X	<p>Only for type Response: text containing status / error code of asynchronous response code for report type or status</p>
response Description	X	O	X	<p>Only for type Response: text containing status / error description of asynchronous response code for report type or status</p>
data	-	O	-	<p>Only for type BinarySms: Contains the binary SMS data as hexadecimal interpretation.</p>
header	-	O	-	<p>Contains the binary header for the SMS as hexadecimal interpretation. This can be used for both text and binary SMSs,</p>



but is normally used only for binary.

Parametr	send	receive	confirm	description
dataCoding Scheme	-	O	-	Contains the data coding scheme of the data fields, supported integer values are 0...255. Normally only used for binary SMS.
protocol Identifier	O	O	-	Protocol Identifier for the SMS message integer values 0...255. This will only appear if the value is different from the default.

#### 5.4.1 Parameter Mappings

This section describes which parameters are required, optional or not supported for each type of object.

Legend:

X: mandatory field

O: optional field

-: field not supported for the type of message

Parameter	TextSms	BinarySms	Response
selector	X	X	X
msgID	X	X	X
refMsgID	-	-	X
fromNumber	X	X	-
toNumber	X	X	-
timestamp	X	X	X
text	X	-	-
responseType	-	-	X
responseCode	-	-	X
responseDescription	-	-	X
data	-	X	-
header	O	O	-

dataCodingScheme	-	X	-
protocolIdentifier	O	O	-

## 5.5 Send

The send action transmits the text SMS with the specified parameters (SMS contents) to the MP. The MP synchronously returns a response indicating a success or an error using responseCode and the responseDescription fields. For asynchronous errors see 3.1. The msgID is returned in the refMsgIDof of the response from the MP; in case the msgID has not been given by the BA, it will contain the message ID generated by the MP.

## 5.6 Receive

The receive action retrieves the first message from the BA's message queue. This may be a text SMS, a binary SMS or a response (MMS is not supported in SMS CONNECTOR). The only input parameter to this action is the baID. If the queue is empty, the MP blocks the response for a defined period of time (deployment specific parameter receiveBlockPeriod; see chapter 11 for more detail). If in the meantime a new message arrives in the queue it is immediately delivered to the BA as a response. After the block period timeout the MP returns an HTTP response status code of 200 (OK) with an empty body.

Depending on the type of received message (field type: TextSMS, BinarySMS, Response) different fields are contained in the contents of the message: for text SMSs the text content is transmitted in the text field; for binary SMSs the data are transmitted using the data, header and dataCodingScheme fields. In case of asynchronous status or error reports, which are received like the other types, the reportText field contains the respective response code.

In case of asynchronous errors the BA will receive (similar to an SMS) a new message of type RESPONSE. The synchronous status will be success (because the message has been received successfully), but the reportText field contains the asynchronous error code.

## 5.7 Confirm

The confirm action must be called after a message has been retrieved from the message platform (i.e. HTTP status code = 200 and the HTTP body is not empty). The action uses the referenced message ID (refMsgID), the referenced BAID (refBaID) and the BA ID (baID) as input parameters to uniquely identifying the message to be confirmed. The MP will consider the message delivered to the BA as soon as it has been confirmed.

If one or more messages are not retrieved by the BA these will be deleted after a certain time period preset by O2 (deployment specific parameter messageReceptionTimeout).

If a message was received but not confirmed within a certain time (deployment specific parameter confirmationTimeout) it is put back on the queue available for polling. Please note that the overall message expiration timeout (deployment specific parameter messageReceptionTimeout) is completely independent of the receive-confirm process. This means that the number of times a BA receives a message but does not confirm it, has no impact on when the message will expire and be deleted from the MP.

The MP supports a limited number of unconfirmed messages and this is defined as a deployment specific parameter, maxUnconfirmedMessages. If this limit is exceeded, the receive action of MP returns a synchronous response with code EAPP\_037 and the BA will not be able to receive messages until the number of unconfirmed messages is reduced below the limit. This can be achieved when the BA issues an explicit confirm or the confirmation timeout expires.

For more information about parallel processing of messages refer to chapter 3.

## 5.8 Examples of Requests and Responses

The following sections show examples of requests and responses using both the GET and POST methods. The target URLs are replaced by placeholders.

### 5.8.1 send

#### 5.8.1.1 GET

##### 5.8.1.2 Request without fromNumber parameter

```
GET /smsconnector/getpost/GP?action=send&baID=1991001&toNumber=%2b420602545687&text=Test+zprava+%3a-
)&msgID=464752c2-4dd4-47f7-8aae-
53053eb5eabc&deliveryReport=TRUE&intruder=FALSE&multipart=FALSE&validityPeriod=10000&priority=1 HTTP/1.1
Connection: Keep-Alive
Host: 127.0.0.1
```

##### 5.8.1.3 Request with fromNumber parameter

```
GET
/smsconnector/getpost/GP?action=send&baID=1991001&fromNumber=%2b420702001001&toNumber=%2b420602545687&text=Test
+zprava+%3a-)&msgID=464752c2-4dd4-47f7-8aae-
53053eb5eabc&deliveryReport=TRUE&intruder=FALSE&multipart=FALSE&validityPeriod=10000&priority=1 HTTP/1.1
Connection: Keep-Alive
Host: 127.0.0.1
```

##### 5.8.1.4 Response

HTTP/1.1 200 OK

```
Server: WebSphere Application Server/5.0
Content-Type: text/plain; charset=utf-8
Content-Language: en-US
Transfer-Encoding: chunked
```

```
ed
selector=Response
responseType=SUCCESS
responseCode=ISUC_001
responseDescription=Send request successfully processed
baID=1991001
refBaID=1991001
msgID=phasdi15z-20040325133509-13
timestamp=2004-03-25T13:39:28
refMsgID=464752c2-4dd4-47f7-8aae-53053eb5eabc
```

0

### 5.8.2 POST

#### 5.8.2.1 Request without fromNumber parameter

```
POST /smsconnector/getpost/GP HTTP/1.1
Content-Type: application/x-www-form-urlencoded
```



Content-Length: 197  
Expect: 100-continue  
Connection: Keep-Alive  
Host: 127.0.0.1

action=send&balD=1991001&toNumber=%2b420602545687&text=Test+zprava+%3a-)&msgID=ceb65a50-1932-4117-95d6-2f6a06d1fa3d&deliveryReport=TRUE&intruder=FALSE&multipart=FALSE&validityPeriod=10000&priority=1

#### 5.8.2.2 Request with fromNumber parameter

POST /smsconnector/getpost/GP HTTP/1.1  
Content-Type: application/x-www-form-urlencoded  
Content-Length: 197  
Expect: 100-continue  
Connection: Keep-Alive  
Host: 127.0.0.1

action=send&balD=1991001 & fromNumber=%2b420702001001&toNumber=%2b420602545687&text=Test+zprava+%3a-)&msgID=ceb65a50-1932-4117-95d6-2f6a06d1fa3d&deliveryReport=TRUE&intruder=FALSE&multipart=FALSE&validityPeriod=10000&priority=1

#### 5.8.2.3 Response

HTTP/1.1 100 Continue

HTTP/1.1 200 OK  
Server: WebSphere Application Server/5.0  
Content-Type: text/plain; charset=utf-8  
Content-Language: en-US  
Transfer-Encoding: chunked

ed  
selector=Response  
responseType=SUCCESS  
responseCode=ISUC\_001  
responseDescription=Send request successfully processed  
balD=1991001  
refBalD=1991001  
msgID=phasdi15z-20040325134534-59  
timestamp=2004-03-25T13:56:14  
refMsgID=ceb65a50-1932-4117-95d6-2f6a06d1fa3d  
0

#### 5.8.3 receive

##### 5.8.3.1 GET

##### 5.8.3.2 Request

GET /smsconnector/getpost/GP?action=receive&balD=1991001 HTTP/1.1  
Host: 10.32.121.87:9999

##### 5.8.3.3 Response







Content-Type: text/plain; charset=utf-8  
 Content-Language: en-US  
 Transfer-Encoding: chunked

```
ee
selector=Response
responseType=SUCCESS
responseCode=ISUC_002
responseDescription=Confirm request successfully processed
baID=1991001
refBaID=1991001
msgID=phasdi15z-20040325134534-62(AppServer)
timestamp=2004-03-25T13:56:20
refMsgID=router@apdevibm4:414005
0
```

## 5.9 Response Types and Codes

The MP returns the status in a synchronous or asynchronous fashion, see also 3.1. Synchronous means the response/ return value of the called action (e.g. response for send action). Asynchronous means that the MP sends a message to the BA's message queue after the originating call has been finished (an example of this would be a delivery report). Asynchronous responses are received in the same way as any other message, such as a text SMS.

There are five types of responses that the messaging platform will return. They are described in the tables below.

Response Type	Value	Synchronous /asynchronous	Definition
Format Error	FORMAT_ERROR	Synchronous	One or more message parameters are in the wrong format
Authentication Error	AUTHORIZATION_ERR OR	Synchronous	An error related to authentication
Application Error	APPL_ERROR	Asynchronous	A negative result of the message processing generated by the Business Application or MP either related to the content of the message or to the functionality.
Internal Error	INTERNAL_ERROR	Synchronous + Asynchronous	A serious error in the O2 Messaging Platform or in the BA in push mode.
Success	SUCCESS	Synchronous + Asynchronous	Synchronous responses for success and asynchronous delivery reports.





The following table presents example contexts for when these response types might happen (while sending of AO or while receiving AT messages) and how the BA may react on these response types.

Response Type	Type	Context/operation Action for BA
Format Error	Send	Display this error to the user to correct the fault of the field (e.g. enter missing field)
Authentication Error	All operations	Check whether BA ID is right and contained in certificate. Check whether certificate has expired or not registered at O2!
Application Error	Send	Display this error to the user to react to the fault, e.g. daily limit exceeded
Internal Error	All operations	Retry action or escalate to O2
Success	Send	Message accepted by MP, message delivery reports

We recommend that the BA does a plausibility check for invalid formats or missing fields on its side before sending a message. This will improve the performance because these basic errors we will be caught immediately instead of waiting for the MP to send back an error response..

For the list of response codes see chapter 9.

## 6 Security

The security of the O2 messaging HTTP GET/POST builds on the use of SSL with client-side certificates. Each BA (identified by its BA ID) will have a client certificate as a statement of its authenticity. For the registration of the BA's certificate (provisioning) at O2 please see the respective documentation provided by O2. This document assumes that the client is already registered in the O2's user registry.

The BA attaches its certificate to the request, which is done by use of the standard mechanisms of the chosen platform. The certificate in the request is then validated at 2 levels:

- The MP checks the validity of the certificate (validity of Certification Authority and verify that the certificate has not expired)
- The MP then checks the authenticity: whether the certificate matches the given BA ID and whether it fits with the one supplied in the provisioning process for this BA (-> authentication)

## 7 Cookies

We recommend that BA accept cookies as we have seen that this may increase performance. It is important that the BA support the "LtpaToken" standard. BAs may connect to the MP without accepting cookies but the performance may be slower.

## 8 Registering a business application

Each application that wants to send and receive SMS using SMS Connector service needs to be registered by O2. Following steps are needed for application registration:

1. Creating certificate – O2 accepts two types of certificates – certificates from public certification authorities (1.CA, Verisign etc.) or certificate generated by O2. If customer wants to use O2 generated certificate, he needs to download from O2 web pages special



application and run it on a local computer. Several company details needs to be specified and application will generate your certificate (public key with extension \*.PEM and private key with extension \*.P12). Additionally, application will print service activation form.

2. Service activation form – O2 contact person in the company needs to fill in service activation form and contact O2 customer care. Additionally, public certificate key needs to be sent by e-mail to customer care. When activating service, customer needs to specify several important details:
  - Voice number to which SMS Connector service will be connected. In case this voice number will be deactivated or suspended, SMS Connector will be deactivated or suspended as well.
  - Used tariff
  - Required capacity limit in number of SMS per second. Standard capacity offered to customer is 1 SMS per second. 5 and 20 SMS per second can be offered for additional fee.
  - Contact person for SMS Connector service – customer contact person, which will be answering inquiries from SMS message recipients
  - Whether customer wants to choose its application number (BAID)
3. Activation of the service – Based on information received from customer O2 will activate the service and will inform customer contact person, which application number has been assigned and what are daily and monthly limits of SMS messages. From this moment customer application will be able to send and receive SMS messages. after 5 seconds. Maximum number of retries should be limited to 5.

## 9 Appendix A: Response Codes

The list of possible response codes with code, description relevant to Business Applications for this interface is attached below, including their relevancy to the given action (Send/Receive/Confirm). O2 reserves the right to change response message texts without notification.



Type	Code	Response message text	Response timing	Send	Receive	Confirm
APPL_ERROR	EAPP_003	BalD není aktivní - EAPP_003 - BalD is not active	Synchronous	X	X	X
APPL_ERROR	EAPP_019	Zpráva zaslána po termínu platnosti zkusebního režimu - EAPP_019 – Trial period has expired	Asynchronous		X	
APPL_ERROR	EAPP_020	Pocet zprav ve zkusebnim rezimu vycerpan - EAPP_020 - No. of messages in trial mode exceeded	Asynchronous		X	
APPL_ERROR	EAPP_023	Prekročen maximální počet zprav, které je možné zaslat v jednom měsíci - EAPP_023 - Maximum quota of messages per month exceeded	Asynchronous		X	
APPL_ERROR	EAPP_024	Prekročen maximální počet zprav, které je možné zaslat během jednoho dne - EAPP_024 - Maximum quota of messages per day exceeded	Asynchronous		X	
APPL_ERROR	EAPP_025	Zpráva potvrzována přes požadavek Confirm není k dispozici - EAPP_025 - No message found for Confirm request	Synchronous			X

Type	Code	Response message text	Response timing	Send	Receive	Confirm
APPL_ERROR	EAPP_037	Limit zpráv čekajících na potvrzení přijetí překročen - EAPP_037 - Too many messages waiting for confirmation	Synchronous	X	X	
APPL_ERROR	EAPP_050	Maximální limit současných požadavků překročen - EAPP_050 - Maximum number of concurrent requests exceeded	Synchronous	X	X	X
APPL_ERROR	EAPP_051	Nadlimitní počet současných požadavků na potvrzení přijetí zpravy - EAPP_051 – Too many concurrent confirm requests	Synchronous			X
APPL_ERROR	EAPP_052	Nadlimitních počet současných požadavků na přijetí zpravy - EAPP_052 - Too many concurrent receive requests	Synchronous		X	
APPL_ERROR	EAPP_053	Nadlimitní počet současných požadavků na zaslání zpravy - EAPP_053 - Too many concurrent send requests	Synchronous	X		
APPL_ERROR	EAPP_060	Aplikace není oprávněna používat číslo MSISDN v poli fromNumber - EAPP_060 - Application is not authorised to use MSISDN number in fromNumber field	Asynchronous		X	
APPL_ERROR	EAPP_101	Zpráva musí být potvrzena přes stejné rozhraní, přes jaké byla přijata - EAPP_101 - Message must be confirmed via same interface through which it was received	Synchronous			X
AUTHORIZATION_ERROR	EAUT_001	Neplatný certifikát - EAUT_001 - Invalid certificate1	Synchronous	X	X	X
<p>1 Business application attempts to establish the https connection with invalid certificate, i.e. with the certificate not provisioned or if the certificate has expired or is suspended.</p>						

Type	Code	Response message text	Response timing	Send	Receive	Confirm
AUTHORIZATION_ERROR	EAUT_002	BA ID není platné nebo není autorizováno pro certifikát - EAUT_002 - BA ID není valid or not authorised for certificate	Synchronous	X	X	X
AUTHORIZATION_ERROR	EAUT_003	Ref BA ID není autorizováno pro certifikát - EAUT_003 - Ref BA ID not authorized for certificate	Synchronous		X	
FORMAT_ERROR	EFMT_004	Neplatný identifikátor zprávy - EFMT_004 - Invalid message ID	Synchronous	X		
FORMAT_ERROR	EFMT_005	Neplatná prioritní zpráva - EFMT_005 - Invalid message priority	Synchronous	X		
FORMAT_ERROR	EFMT_006	Nesouhlasí čísla fromNumber s BA ID - EFMT_006 - Invalid fromNumber (mismatch with BA ID)	Synchronous	X		
FORMAT_ERROR	EFMT_007	Neplatný formát MSISDN čísla příjemce - EFMT_007 - Invalid recipient MSISDN number format	Synchronous	X		
FORMAT_ERROR	EFMT_009	Příliš dlouhý suffix zprávy - EFMT_009 - Message suffix too long	Synchronous	X		
FORMAT_ERROR	EFMT_010	Neplatný formát MSISDN čísla fromNumber - EFMT_010 - Invalid MSISDN number format of fromNumber	Synchronous	X		
FORMAT_ERROR	EFMT_011	Suffix zprávy musí být číselná hodnota - EFMT_011 - Message suffix must be numeric	Synchronous	X		
FORMAT_ERROR	EFMT_012	Jen jedno z poli suffix a fromNumber může být vyplněno - EFMT_012 - Only one of the suffix and fromNumber fields might be filled in	Synchronous	X		
FORMAT_ERROR	EFMT_013	Neplatná hodnota pole Report Level / Delivery Report - EFMT_013 - Invalid Report Level / Delivery Report	Synchronous	X		
FORMAT_ERROR	EFMT_016	Neplatný formát pole Validity period - EFMT-016 - Incorrect message validity	Synchronous	X		

Type	Code	Response message text	Response timing	Send	Receive	Confirm
FORMAT_ERROR	EFMT_025	SMS nemoze byt soucasne poslana jako multipart a intruder - EFMT_025 - SMS cannot be sent as multipart and intruder at the same time.	Synchronous	X		
FORMAT_ERROR	EFMT_026	Nepripustna delka zpravy - EFMT_026 - Message too long	Synchronous	X		
FORMAT_ERROR	EFMT_032	Pole BA ID je prazdne - EFMT_032 - BA ID empty	Synchronous	X	X	X
FORMAT_ERROR	EFMT_034	Zprava neobsahuje zadny text - EFMT_034 – Text message empty	Synchronous	X		
FORMAT_ERROR	EFMT_035	Neplatny pozadavek - EFMT_035 - Bad Request	Synchronous	X	X	X
FORMAT_ERROR	EFMT_101	Neplatna hodnota pole Intruder - EFMT_101 - Invalid intruder format	Synchronous	X		
FORMAT_ERROR	EFMT_102	Neplatna hodnota pole multipart - EFMT_102 - Invalid multipart format	Synchronous	X		
FORMAT_ERROR	EFMT_105	Text zpravy obsahuje nepodporovane znaky - EFMT_105 - SMS text contains unsupported characters	Synchronous	X		
FORMAT_ERROR	EFMT_106	Neplatna hodnota pole data coding scheme - EFMT_106 - Invalid data coding scheme	Synchronous	X		
FORMAT_ERROR	EFMT_121	Neplatna hodnota pole protocolIdentifier - EFMT_121 - Invalid protocolIdentifier value	Synchronous	X		
INTERNAL_ERROR	EINT_005	Zprava zamitnuta SMSC/MMSC - EINT_005 - Message rejected by SMSC/MMSC2	Asynchronous		X	
INTERNAL_ERROR	EINT_012	O2: Chyba - Zprava nedorucena z technickyh duvodu - EINT_012 - Message not delivered for technical reasons	Synchronous	X	X	X
2. Most probable reason of EINT_005 is incorrect international MSISDN prefix or incorrect MSISDN length.						

Type	Code	Response message text	Response timing	Send	Receive	Confirm
INTERNAL_ERROR	EINT_102	Zprava nedorucena z technickych duvodu - EINT_102 - Message not delivered for technical reasons	Asynchronous	X		
SUCCESS	ISUC_001	Pozadavek Send uspesne zpracovan - ISUC_001 - Send request successfully processed	Synchronous	X		
SUCCESS	ISUC_002	Pozadavek Confirm uspesne zpracovan - ISUC_002 - Confirm request successfully processed	Synchronous			X
SUCCESS	ISUC_005	Zprava byla dorucena - ISUC_005 - Message delivered	Asynchronous		X	
SUCCESS	ISUC_006	Zprava nebyla dorucena - ISUC_006 - Message delivery failed	Asynchronous		X	
SUCCESS	ISUC_010	Zprava byla dorucena na SMSC/MMSC - ISUC_010 - Message forwarded to SMSC/MMSC	Asynchronous		X	



## 10 Appendix B: response codes returned to end-user's mobile phone

The following table describes the list of response codes, which are generated as a result of MO SMS sent from end-user's mobile phone, and which are returned by the MP in case of error in the form of informative SMS to end-user's mobile (please see the note below).



Type	Code	Response message text
APPL_ERROR	EAPP_033	O2: Chyba - Prilis dlouhe cislo sluzby - EAPP_033 - Service number too long
APPL_ERROR	EAPP_034	O2: Chyba - Cislo sluzby neexistuje – EAPP_034 - Service number does not exist
APPL_ERROR	EAPP_047	O2: Chyba - Zprava nedorucena - casovy limit prekrocen - EAPP_047 - Message not delivered - timeout expired
FORMAT_ERROR	EAPP_113	O2: Chyba - Zprava nedorucena z technickych duvodu - EAPP_113 - Message not delivered for technical reasons
INTERNAL_ERROR	EINT_012	O2: Chyba - Zprava nedorucena z technickych duvodu - EINT_012 - Message not delivered for technical reasons
INTERNAL_ERROR	EINT_107	O2: Chyba - Zprava nedorucena z technickych duvodu - EINT_107 - Message not delivered for technical reasons
INTERNAL_ERROR	EINT_108	O2: Chyba - Zprava nedorucena z technickych duvodu - EINT_108 - Message not delivered for technical reasons
INTERNAL_ERROR	EINT_109	O2: Chyba - Zprava nedorucena z technickych duvodu - EINT_109 - Message not delivered for technical reasons

## 11 Appendix C: Deployment Specific Parameters

Please see the document “SMS Connector Deployment Specific Parameters”.

## 12 Appendix D: Solution constraints

The following lists known MP constraints:

- Maximum value of validityPeriod used for SMS AO message will be 388800 seconds (4,5 days), even if BA sets higher value of validityPeriod field
- Maximum time, during which the platform sends Delivery Reports (positive or negative - ISUC\_005 and ISUC\_006 Response codes) as responses to AO SMS messages is 46800 seconds (13 hours). If BA asks for Delivery reports, sets validityPeriod SMS field value higher and mobile terminal is not reached within 13 hours, neither positive nor negative Delivery Report will be sent back to BA.
- In exceptional circumstances the BA can receive an HTTP error 403 request, even if there is no problem with authorization. The BA should repeat the same request after 5 seconds. Maximum number of retries should be limited to 5.

