

Business SMS

SMS Gateway | XML Interface

Stand: 12/2008



sms.ch AG | a Buongiorno Company
CH-9470 Buchs | Technikumstrasse 14
T: +41 81 599 71 00
F: +41 81 599 71 01
kontakt@business.sms.ch | business.sms.ch

Handelsregister Nr.: CH-514.3.025.232-4
Kanton St. Gallen
MwSt. Nr.: 524 218

VERSION

Version	Autor	Date	Remarks
0.1	Martin Mrvka	27.07.2004	Draft
0.2	Martin Mrvka	20.08.2004	Added Error Codes
0.3	Gerd Reifenauer	08.09.2004	Added AutoSegment Option ,none'
0.4	Gerd Reifenauer	21.10.2004	Added MO/COD-Specs
0.5	Gerd Reifenauer	11.03.2005	Added Error Code
0.6	Gerd Reifenauer	12.05.2005	Added attributes message Priority and cod ReturnAddress
0.7	Gerd Reifenauer	28.06.2006	Added attribute OperatorID in MO Forward Sender
0.71	Gerd Reifenauer	12.01.2007	Added links to DTD Files
0.8	Martin Mrvka	04.12.2008	Added support for WAP Push

TABLE OF CONTENT

Version	I
1 Introduction	1
2 Purpose of this Document	1
3 Terms of use	1
4 Interface Specification	1
4.1 System Overview.....	1
4.2 Interface Location.....	1
4.3 Interface Access	1
4.4 Sendig MT Messages	2
4.4.1 XML Request Definition	2
4.4.2 XML Response Definition	5
4.5 Receiving COD Messages	6
4.5.1 XML Report Definition	6
4.5.2 XML Report Response Definition.....	7
4.6 Receiving MO Messages	8
4.6.1 XML Forward Definition	8
4.6.2 XML Forward Response Definition.....	10
Document Type Definitions	11
4.7 MT Request DTD.....	11
4.8 MT Response DTD.....	11
4.9 COD Request DTD.....	12
4.10 COD Response DTD.....	12
4.11 MO Forward DTD	12
4.12 MO Forward Response DTD	12
4.13 Error Codes.....	13
4.14 DTD Files online.....	13
5 Examples	14
5.1 MT Request.....	14
5.2 MT Response Error.....	14
5.3 MT Response Success.....	14

5.4	Report.....	14
5.5	Report Response Error.....	15
5.6	Report Response Success	15
5.7	Forward.....	15
5.8	Forward Response Error	15
5.9	Forward Response Success	16
5.10	WapPush Request	16
5.11	WapPush Response Error	16
5.12	WapPush Response Success	16

1 INTRODUCTION

sms.at operates a sms gateway system which is directly connected to all Austrian mobile network operators and international sms carriers.

2 PURPOSE OF THIS DOCUMENT

This document describes the interface provided to third parties requiring connectivity to the sms.at sms gateway system.

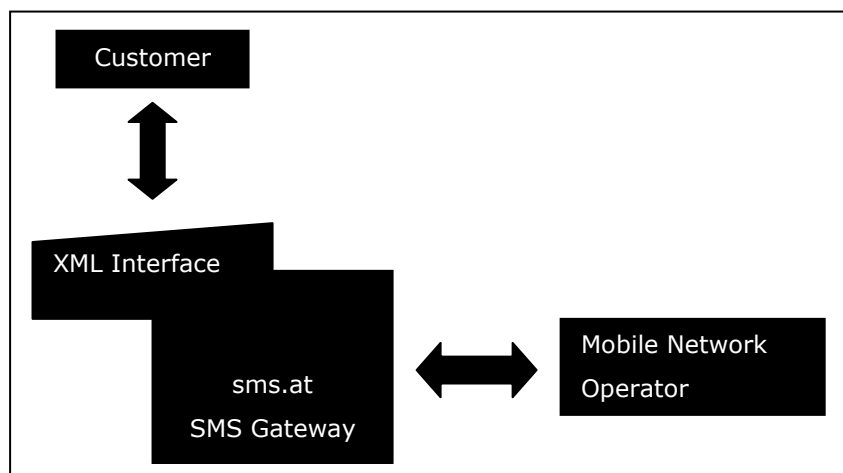
It is intended for development staff to attach the third parties infrastructure in order to send sms messages to the mobile network operators and carriers connected to the sms.at sms gateway system.

3 TERMS OF USE

No part of this document may be disclosed orally or in writing, including by reproduction, to any third party without the prior written consent sms.at mobile internet services gmbh. This document, its associated appendices and any attachments remain the property of sms.at and shall be returned upon request.

4 INTERFACE SPECIFICATION

4.1 SYSTEM OVERVIEW



4.2 INTERFACE LOCATION

The Interface is located at:
http://gateway.sms.at/xml_interface/

4.3 INTERFACE ACCESS

The interface is used by exchanging XML documents. The transport of these XML documents is done by HTTP POST method. The transport of XML documents is initiated in two different ways, depending on the document type.

For sending MT Messages the connection is initiated by the customer to the interface, for receiving COD Reports or MO Messages it is initiated by the sms.at interface to the customer.

(Reports and MO Messages can also be received by email)

4.4 SENDIG MT MESSAGES

A XML document containing one Request must be sent to the interface. Usually one XML document describes a SMS message for one recipient. In this case the Message is directly forwarded to the mobile network operator.

Documents for MT Messages can hold one or more message parts for multiple recipients. If more than one recipients are given within the same request, the interface will store the request for deferred delivery. The interface will answer the document transfer with a Response XML document.

4.4.1 XML REQUEST DEFINITION

Please see chapter MT Request DTD for complete DTD.

4.4.1.1 REQUEST

```
<!ELEMENT Request (AccountLogin, AccountPass, Message)>
```

The Request element is used to encapsulate the message. Each request is associated to an account. The account is identified by a login and a password. The login may be the accounts username or the assigned email address.

4.4.1.2 ACCOUNT LOGIN

```
<!ELEMENT AccountLogin (#PCDATA)>  
<!ATTLIST AccountLogin  
  Type (Email | User) #REQUIRED  
>
```

The AccountLogin identifies an account. With the attribute Type, it is defined if the AccountLogin holds an email address or a username.

4.4.1.3 ACCOUNT PASS

```
<!ELEMENT AccountPass (#PCDATA)>
```

The AccountPass verifies the given login.

4.4.1.4 MESSAGE

```
<!ELEMENT Message (Sender? | Recipients | (Data | Text) | Cod?)>
<!ATTLIST Message
  Type CDATA #REQUIRED
  Alphabet CDATA #IMPLIED
  Class CDATA #IMPLIED
  Id CDATA #IMPLIED
  Priority CDATA #IMPLIED
>
```

The Message element is used to carry the message itself. A Message is must be identified with a specific type.

Attribute	Description
Type	MTSMS ... SMS Message delivered to a mobile device
ID	Optional ID for the message, will be returned with COD
Alphabet	0 ... default text encoding (ISO-8859-1) 1 ... 8bit binary encoding 2 ... 16bit UCS2 encoding
Class	0 ... don't write SMS to SIM (Flash Message) 1 ... write SMS to SIM (default)
Priority	0 ... default

4.4.1.5 RECIPIENTS

```
<!ELEMENT Recipients (Recipient+)>
```

Within the Recipients element, a list for recipient addresses for this Message can be defined. The must be at least one recipient address.

4.4.1.6 RECIPIENT

```
<!ELEMENT Recipient (#PCDATA)>
<!ATTLIST Recipient
  Type (RFC822 | International) #REQUIRED
  Id CDATA #IMPLIED
>
```

The Recipient identifies the recipient's address this can either be a mobile number (e.g. for SMS requests) or an email address (e.g. for SMTP forwarding).

Attribute	Description
Type	RFC822 ... Email Address International ... International MSISDN (eg. 436761234567)
Id	Optional ID for each recipient, will be returned with COD

4.4.1.7 DATA

```
<!ELEMENT Data (Segment+)>
```

The Data Element is used to send the different types of binary sms messages. It may contain one or more message segments (for concatenated messages).

4.4.1.8 SEGMENT

```
<!ELEMENT Segment (#PCDATA)>
<!ATTLIST Segment
    UDH CDATA #IMPLIED
>
```

Within a Segment Element, the content of a message part can be defined. The content has to be hex coded. Additionally the attribute UDH may define the hex coded User Data Header.

Attribute	Description
UDH	Hex coded user data header

4.4.1.9 TEXT

```
<!ELEMENT Text (#PCDATA)>
<!ATTLIST Text
    AutoSegment (none | simple | 8bitref | 16bitref) #REQUIRED
>
```

The Text Element is used to transmit simple text messages. Since a SMS message may not contain more than 160 characters, the text can be split up by the interface into the proper parts. The message characters need to be hex coded.

Attribute	Description
AutoSegment	none ... don't split, send a maximum of 160 characters simple ... split into parts with 160 characters each 8bitref ... split into parts using an 8bit reference number 16bitref ... split into parts using an 16bit reference number

4.4.1.10 WAPPUSH

```
<!ELEMENT Text (#PCDATA)>
<!ATTLIST Text
    AutoSegment (none | simple | 8bitref | 16bitref) #REQUIRED
>
```

The WapPush Element is used to transmit wap push messages in an easy way. The message characters need to be hex coded.

Currently the most common used type of a wap push message "Service Indication" is supported.

As a rule of thumb, the length of the URL together with the message should not exceed 120 characters to fit into a single SMS.

Attribute	Description
Type	The type of the wap push message SI ... Service Indication
Url	The full qualified URL (e.g. http://wap.example.com)
AllowMultiSegment	Explicitly allow to generate multiple SMS when the URL and the text don't fit into a single SMS.

4.4.1.11 Cod

```
<!ELEMENT Cod (#PCDATA)>
<!ATTLIST Cod
    ReturnAddress #IMPLIED
>
```

The Cod Element is used to enable Delivery Notification for the given recipients if available. Content can be 0 or 1.

Attribute	Description
ReturnAddress	Return Address for COD Report Can be common mail-address e.G.: "name@host.com" or http-address e.G.: "http://my.host.com/receive_report/" Overrules account settings.

4.4.1.12 SENDER

```
<!ELEMENT Sender (#PCDATA)>
<!ATTLIST Sender
    Type (RFC822 | Shortcode | National | International | Alphanumeric) #REQUIRED
>
```

The Sender identifies the sender's address this can either be a mobile number (e.g. for SMS requests) or an email address (e.g. for SMTP forwarding).

Attribute	Description
Type	RFC822 ... Email Address International ... International MSISDN (eg. 436761234567) Alphanumeric 11 chars, Latin1 encoding

Due to restrictions in the mobile network infrastructure, it is not allowed to set Sender completely free. Please contact the key account to get your available list for sending addresses.

4.4.2 XML RESPONSE DEFINITION

Please see chapter [MT Response DTD](#) for complete DTD.

4.4.2.1 RESPONSE

```
<!ELEMENT Response Code,CodeDescription,TransferId?>
```

The Response element is used to encapsulate the response message.

4.4.2.2 CODE

```
<!ELEMENT Code (#PCDATA)>
```

The Code Element is used to represent the different return values. Please see chapter [Error Codes](#) for the detailed return values.

4.4.2.3 CODEDESCRIPTION

```
<!ELEMENT CodeDescription (#PCDATA)>
```

The CodeDescription Element gives more information for a returned code. Please see chapter [Error Codes](#) for the detailed description.

4.4.2.4 TRANSFERID

```
<!ELEMENT TransferId (#PCDATA)>
```

If a request could be successfully processed, the TransferId Element holds a unique ID.

4.5 RECEIVING COD MESSAGES

The sms.at interface will send each delivery report (COD) as XML document containing one [Report](#) to the customer's assigned URL.

XML Documents are transferred by HTTP POST method.

The Report has to be answered with a document containing [ReportResponse](#). Without positive response, the sms.at interface will retry to send the Report for a certain amount of time.

4.5.1 XML REPORT DEFINITION

Please see chapter [COD Request DTD](#) for complete DTD.

4.5.1.1 REPORT

```
<!ELEMENT Report (AccountLogin, AccountPass, COD)>
```

The Report element is used to encapsulate the confirmation of delivery (COD). Each report transfer request is associated to an account. The account is identified by a login and a password.

4.5.1.2 ACCOUNTLOGIN

```
<!ELEMENT AccountLogin (#PCDATA)>
```

The AccountLogin identifies an account. The AccountLogin holds an username.

4.5.1.3 ACCOUNTPASS

```
<!ELEMENT AccountPass (#PCDATA)>
```

The AccountPass verifies the given login

4.5.1.4 COD (CONFIRMATION OF DELIVERY)

```
<!ELEMENT COD (#PCDATA)>
<!ATTLIST COD
  TransferID CDATA #REQUIRED
  SenderAddress CDATA #REQUIRED
  Status CDATA #REQUIRED
  SentOn CDATA #IMPLIED
  DeliveredOn CDATA #IMPLIED
  MessageID CDATA #IMPLIED
  RecipientID CDATA #IMPLIED
>
```

The COD element is used to carry the confirmation of delivery itself and has at least 3 obligate attributes (TransferID,SenderAddress,Status).

Attribute	Description
TransferId	COD belongs to Send Request with this TransferId.
SenderAddress	Mobile number this COD belongs to
Status	delivered ... Message was delivered to SenderAddress undelivered ... Message was not delivered expired ... Message validity expired deleted ... Message was deleted rejected ... Message was not accepted for Sender
SentOn	Date and time when message was sent to SenderAddress
DeliveredOn	Date and time when SenderAddress returned COD status
MessageID	Optional ID for the message, was given with send request
RecipientID	Optional ID for this recipient, was given with send request

4.5.2 XML REPORT RESPONSE DEFINITION

Please see chapter [COD Response DTD](#) for complete DTD.

4.5.2.1 REPORT RESPONSE

```
<!ELEMENT ReportResponse (Code, CodeDescription?)>
```

The ReportResponse element is used to encapsulate the response message after Report transfer.

4.5.2.2 CODE

```
<!ELEMENT Code (#PCDATA)>
```

The Code Element is used to represent the different return values. Please see chapter [Error Codes](#) for the detailed return values.

4.5.2.3 CODEDESCRIPTION

```
<!ELEMENT CodeDescription (#PCDATA)>
```

Optional.

The CodeDescription Element gives more information for a returned code. Please see chapter [Error Codes](#) for the detailed description.

4.6 RECEIVING MO MESSAGES

The sms.at interface will send each MO Message as XML document containing one [Forward](#) to the customer's assigned URL. XML documents are transferred by HTTP POST method. The Forward has to be answered with a document containing ForwardResponse. Without positive response, the sms.at interface will retry to send the MO Message for a certain amount of time.

4.6.1 XML FORWARD DEFINITION

Please see chapter [MO Forward DTD](#) for complete DTD.

4.6.1.1 FORWARD

```
<!ELEMENT Forward (AccountLogin, AccountPass, Message)>
```

The Forward element is used to encapsulate the mobile originated message (MO). Each Forward request is associated to an account. The account is identified by a login and a password.

4.6.1.2 ACCOUNTLOGIN

```
<!ELEMENT AccountLogin (#PCDATA)>
```

The AccountLogin identifies an account. The AccountLogin holds an username.

4.6.1.3 ACCOUNTPASS

```
<!ELEMENT AccountPass (#PCDATA)>
```

The AccountPass verifies the given login.

4.6.1.4 MESSAGE

```
<!ELEMENT Message (Sender, Recipient, Data)>
```

The Message element is used to carry the message itself. A Message is identified with a specific type.

Attribute	Description
Type	MOSMS ... SMS Message delivered (forwarded) from a mobile device
Alphabet	0 ... default text encoding (ISO-8859-1) 1 ... 8bit binary encoding 2 ... 16bit UCS2 encoding
Class	0 ... don't write SMS to SIM (Flash Message) 1 ... write SMS to SIM (default)

4.6.1.5 SENDER

```
<!ELEMENT Sender (#PCDATA)>
<!ATTLIST Sender
  Type (RFC822 | International) #REQUIRED
  OperatorID CDATA #IMPLIED
>
```

The Sender identifies the sender's address this can either be a mobile number (e.g. for SMS requests) or an email address (e.g. for SMTP forwarding).

Attribute	Description
Type	RFC822 ... Email Address International ... International MSISDN (eg. 436761234567)
OperatorID	ID of network operator over which message was sent. (eg. 2) Optional – Attribute only given when available. XML List of used Operator IDs can be found at http://gateway.sms.at/xml/interface/OperatorID/

4.6.1.6 RECIPIENT

```
<!ELEMENT Recipient (#PCDATA)>
<!ATTLIST Recipient
  Type (RFC822 | International | National | Shortcode | Alphanumeric) #REQUIRED
>
```

The Recipient identifies the recipient address of this MO message forward. (Usually a customer's assigned sender address for mt messages). It can either be a number or an email address (e.g. for SMTP forwarding).

Attribute	Description
Type	RFC822 ... Email Address International ... International MSISDN (eg. 436761234567) National ... National notation (eg. 06761234567) Shortcode ... Network internal shortcode (eg. 12345) Alphanumeric ... Max. 11 Char alphanumeric string

4.6.1.7 DATA

```
<!ELEMENT Data (Segment+)>
```

The Data Element is used to send the single segments of the binary or text message. It may contain one or more message segments (for concatenated messages).

Data with multiple message segments of text (not binary) can be sent as one part within the [Text](#) Element for easier handling.

4.6.1.8 SEGMENT

```
<!ELEMENT Segment (#PCDATA)>
<!ATTLIST Segment
    UDH CDATA #IMPLIED
>
```

Within a Segment Element, the content of a message part is defined. The content has to be hex coded. Additionally the attribute UDH may define the hex coded User Data Header.

Attribute	Description
UDH	Hex coded user data header

4.6.1.9 TEXT

```
<!ELEMENT Text (#PCDATA)>
```

Content is hex coded. The Text Element is added to documents containing Data of multiple text message segments. It holds the concatenated text message.

4.6.2 XML FORWARD RESPONSE DEFINITION

Please see chapter [MO Forward Response DTD](#) for complete DTD.

4.6.2.1 FORWARD RESPONSE

```
<!ELEMENT ForwardResponse (Code, CodeDescription?)>
```

The ForwardResponse element is used to encapsulate the response message after MO message transfer.

4.6.2.2 CODE

```
<!ELEMENT Code (#PCDATA)>
```

The Code Element is used to represent the different return values. Please see chapter [Error Codes](#) for the detailed return values.

4.6.2.3 CODEDESCRIPTION

```
<!ELEMENT CodeDescription (#PCDATA)>
```

Optional. The CodeDescription Element gives more information for a returned code. Please see chapter [Error Codes](#) for the detailed description.

DOCUMENT TYPE DEFINITIONS

4.7 MT REQUEST DTD

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!ELEMENT Request (AccountLogin, AccountPass, Message)>
<!ELEMENT AccountLogin (#PCDATA)>
<!ATTLIST AccountLogin
    Type (Email | User) #REQUIRED
>
<!ELEMENT AccountPass (#PCDATA)>
<!ELEMENT Message (Sender? , Recipients , (Data | Text) , Cod?)>
<!ATTLIST Message
    Type CDATA #REQUIRED
    Alphabet CDATA #IMPLIED
    Class CDATA #IMPLIED
    Id CDATA #IMPLIED
    Priority CDATA #IMPLIED
>
<!ELEMENT Sender (#PCDATA)>
<!ATTLIST Sender
    Type (RFC822 | Shortcode | National | International | Alphanumeric) #REQUIRED
>
<!ELEMENT Recipients (Recipient+)>
<!ELEMENT Recipient (#PCDATA)>
<!ATTLIST Recipient
    Type (RFC822 | International) #REQUIRED
    Id CDATA #IMPLIED
>
<!ELEMENT Data (Segment+)>
<!ELEMENT Segment (#PCDATA)>
<!ATTLIST Segment
    UDH CDATA #IMPLIED
>
<!ELEMENT Text (#PCDATA)>
<!ATTLIST Text
    AutoSegment (none | simple | 8bitref | 16bitref) #REQUIRED
>
<!ELEMENT WapPush (#PCDATA)>
<!ATTLIST Text
    Type (SI) #REQUIRED
    Url (#PCDATA) #REQUIRED
    AllowMultiSegment (yes|no)
>
<!ELEMENT Cod (#PCDATA)>
<!ATTLIST Cod
    ReturnAddress CDATA #IMPLIED
>
```

4.8 MT RESPONSE DTD

```
<?xml version="1.0" encoding="ISO-8859-1"?>
< !ELEMENT Response (Code, CodeDescription, TransferId ?)>
< !ELEMENT Code (#PCDATA)>
< !ELEMENT CodeDescription (#PCDATA)>
< !ELEMENT TransferId (#PCDATA)>
```

4.9 COD REQUEST DTD

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!ELEMENT Report (AccountLogin, AccountPass, COD)>
<!ELEMENT AccountLogin (#PCDATA)>
<!ELEMENT AccountPass (#PCDATA)>
<!ELEMENT COD (#PCDATA)>
<!ATTLIST COD
  TransferID CDATA #REQUIRED
  SenderAddress CDATA #REQUIRED
  Status CDATA #REQUIRED
  SentOn CDATA #IMPLIED
  DeliveredOn CDATA #IMPLIED
  MessageID CDATA #IMPLIED
  RecipientID CDATA #IMPLIED
>
```

4.10 COD RESPONSE DTD

```
< ?xml version= »1.0 » encoding= »ISO-8859-1 » ?>
<!ELEMENT ReportResponse (Code, CodeDescription?)>
<!ELEMENT Code (#PCDATA)>
<!ELEMENT CodeDescription (#PCDATA)>
```

4.11 MO FORWARD DTD

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!ELEMENT Forward (AccountLogin, AccountPass, Message)>
<!ELEMENT AccountLogin (#PCDATA)>
<!ELEMENT AccountPass (#PCDATA)>
<!ELEMENT Message (Sender, Recipient, Data, Text?)>
<!ATTLIST Message
  Type CDATA #REQUIRED
  Alphabet CDATA #IMPLIED
  Class CDATA #IMPLIED
>
<!ELEMENT Sender (#PCDATA)>
<!ATTLIST Sender
  Type (RFC822 | International) #REQUIRED
  OperatorID CDATA #IMPLIED
>
<!ELEMENT Recipient (#PCDATA)>
<!ATTLIST Recipient
  Type (RFC822 | International | National | Shortcode | Alphanumeric) #REQUIRED
>
<!ELEMENT Data (Segment+)>
<!ELEMENT Segment (#PCDATA)>
<!ATTLIST Segment
  UDH CDATA #IMPLIED
>
<!ELEMENT Text (#PCDATA)>
```

4.12 MO FORWARD RESPONSE DTD

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!ELEMENT ForwardResponse (Code, CodeDescription)>
<!ELEMENT Code (#PCDATA)>
<!ELEMENT CodeDescription (#PCDATA)>
```

4.13 ERROR CODES

Code	CodeDescription	Remarks
2000	OK - ACCEPTED	The request has been accepted by the network operator
2001	OK - QUEUED	The request has been queued for delivery
4000	ERR - BAD XML	The XML document could not be parsed
4001	ERR - INVALID CREDENTIALS	The access is denied due to wrong login data
4002	ERR - INVALID RECIPIENTS	The given recipients are invalid
4003	ERR - INVALID SENDER	The given sender is invalid
4004	ERR - INVALID MESSAGE TYPE	The given message type is invalid
4005	ERR - INVALID ALPHABET	The given alphabet value is invalid
4006	ERR - INVALID CLASS	The given message class is invalid
4007	ERR - INVALID DATA	One of the message segments is invalid
4008	ERR - INVALID MESSAGE ID	The given message id is invalid
4009	ERR - INVALID TEXT	The text part contains errors
4010	ERR - INVALID AUTOSEGMENT	The given autosegment value is invalid
4011	ERR - INVALID COD	The given cod value is invalid
4012	ERR - THROTTLING	The transfer rate for immediate transmissions exceeded
4013	ERR - MSG LIMIT EXCEEDED	The allowed message limit is hit
4014	ERR - UNAUTHORIZED IP	Sender IP address is not authorized for login
4015	ERR - INVALID MESSAGE PRIORITY	The given message priority value is invalid
4016	ERR - INVALID COD RETURN ADDRESS	The given cod ReturnAddress is invalid.
4017	ERR - MULTISEGMENTS	The value for AllowMultiSegments is invalid
5000	ERR - INTERNAL ERROR	An internal system error occurred

4.14 DTD FILES ONLINE

http://gateway.sms.at/xml_interface/dtd/outlooksms_mt_request.dtd
http://gateway.sms.at/xml_interface/dtd/outlooksms_mt_response.dtd
http://gateway.sms.at/xml_interface/dtd/outlooksms_report.dtd
http://gateway.sms.at/xml_interface/dtd/outlooksms_report_response.dtd
http://gateway.sms.at/xml_interface/dtd/outlooksms_mo_forward.dtd
http://gateway.sms.at/xml_interface/dtd/outlooksms_mo_forward_response.dtd

5 EXAMPLES

5.1 MT REQUEST

```
POST /xml_interface/ HTTP/1.0
Content-Type: text/xml
Host: gateway.sms.at
Content-Length: 369

<?xml version="1.0" encoding="ISO-8859-1"?>
<Request>
  <AccountLogin Type="email">test@sms.at</AccountLogin>
  <AccountPass>test</AccountPass>
  <Message Type="MTSMS" Id="1">
    <Recipients>
      <Recipient Type="International" Id="1">436761234567</Recipient>
    </Recipients>
    <Text AutoSegment="simple">74657374</Text>
    <Cod>0</Cod>
  </Message>
</Request>
```

5.2 MT RESPONSE ERROR

```
HTTP/1.1 200 OK
Content-Type: text/xml

<?xml version='1.0' encoding='ISO-8859-1'?>
<Response>
  <Code>4001</Code>
  <CodeDescription>ERR - INVALID CREDENTIALS</CodeDescription>
</Response>
```

5.3 MT RESPONSE SUCCESS

```
HTTP/1.1 200 OK
Content-Type: text/xml

<?xml version='1.0' encoding='ISO-8859-1'?>
<Response>
  <Code>2000</Code>
  <CodeDescription>OK - ACCEPTED</CodeDescription>
  <TransferId>004125f9d9000000007e</TransferId>
</Response>
```

5.4 REPORT

```
POST /customer_url/ HTTP/1.0
Content-Type: text/xml
Host: customer.com
Content-Length: 369

<?xml version="1.0" encoding="ISO-8859-1"?>
<Report>
  <AccountLogin>user</AccountLogin>
  <AccountPass>pass</AccountPass>
  <COD TransferID='004125f9d9000000007f'
    SenderAddress='436761234567'
    Status='delivered'
    SentOn='2004-09-29 08:45:00'
    DeliveredOn='2004-09-30 12:00:00'
    MessageID='1test'
  >
```

```

    RecipientID='herrtest'
  />
</Report>

```

5.5 REPORT RESPONSE ERROR

```

HTTP/1.1 200 OK
Content-Type: text/xml

<?xml version='1.0' encoding='ISO-8859-1'?>
<ReportResponse>
  <Code>4001</Code>
  <CodeDescription>ERR - INVALID CREDENTIALS</CodeDescription>
</ReportResponse>

```

5.6 REPORT RESPONSE SUCCESS

```

HTTP/1.1 200 OK
Content-Type: text/xml

<?xml version='1.0' encoding='ISO-8859-1'?>
<ReportResponse>
  <Code>2000</Code>
</ReportResponse>

```

5.7 FORWARD

```

POST /customer_url/ HTTP/1.0
Content-Type: text/xml
Host: customer.com
Content-Length: 487

<?xml version='1.0' encoding='ISO-8859-1'?>
<Forward>
  <AccountLogin>user</AccountLogin>
  <AccountPass>pass</AccountPass>
  <Message Type='MOSMS' Alphabet='1' Class='0'>
    <Sender Type='International'>436766688656</Sender>
    <Recipient Type='National'>08282707000004</Recipient>
    <Data>
      <Segment>44696573206973742065696e6520546578746e61636872696368742e</Segment>
    </Data>
    <Text>44696573206973742065696e6520546578746e61636872696368742e </Text>
  </Message>
</Forward>

```

5.8 FORWARD RESPONSE ERROR

```

HTTP/1.1 200 OK
Content-Type: text/xml

<?xml version='1.0' encoding='ISO-8859-1'?>
<ForwardResponse>
  <Code>4001</Code>
  <CodeDescription>ERR - INVALID CREDENTIALS</CodeDescription>
</ForwardResponse>

```

5.9 FORWARD RESPONSE SUCCESS

```
HTTP/1.1 200 OK
Content-Type: text/xml

<?xml version='1.0' encoding='ISO-8859-1'?>
<ForwardResponse>
  <Code>2000</Code>
</ForwardResponse>
```

5.10 WAPPUSH REQUEST

```
POST /xml_interface/ HTTP/1.0
Content-Type: text/xml
Host: gateway.sms.at
Content-Length: 369

<?xml version="1.0" encoding="ISO-8859-1"?>
<Request>
  <AccountLogin Type="email">test@sms.at</AccountLogin>
  <AccountPass>test</AccountPass>
  <Message Type="MTSMS" Id="1">
    <Recipients>
      <Recipient Type="International" Id="1">436761234567</Recipient>
    </Recipients>
    <WapPush Type="SI" Url="http://wap.test.at">74657374</Text>
    <Cod>0</Cod>
  </Message>
</Request>
```

5.11 WAPPUSH RESPONSE ERROR

```
HTTP/1.1 200 OK
Content-Type: text/xml

<?xml version='1.0' encoding='ISO-8859-1'?>
<Response>
  <Code>4001</Code>
  <CodeDescription>ERR - INVALID CREDENTIALS</CodeDescription>
</Response>
```

5.12 WAPPUSH RESPONSE SUCCESS

```
HTTP/1.1 200 OK
Content-Type: text/xml

<?xml version='1.0' encoding='ISO-8859-1'?>
<Response>
  <Code>2000</Code>
  <CodeDescription>OK - ACCEPTED</CodeDescription>
  <TransferId>004125f9d9000000007e</TransferId>
</Response>
```