

Send SMS HTTP Specification

Document Version: 0.2

Date: 25-02-2011

This document explains the HTTP Specification for sending an SMS through the Directologies Enterprise Messaging Suite.

Table Of Contents

Sr. No.	Section	Page No.
1	Document Revision History	3
2	Introduction	4
3	HTTP URL	4
4	HTTP POST Parameters	4
5	Success Response	5
6	Error Response	5
7	Errors	6
8	Examples	7

1. Document Revision History:

Version 0.1: 23-February-2011

- Initial Version.

Version 0.2: 25-February-2011

- Added Error for Destination Format which was previously reported under URL Format error.

2. Introduction:

This document describes the HTTP Specification for the SMS Send Service from Direct Technologies Pvt. Ltd. It details the names of the parameters to be used in the HTTP POST method and also mentions possible values for them. It also mentions the success and error response format, various error codes and messages and possible resolutions for them.

This document would be required by someone, most probably a developer or a program designer, who wishes to write an application that can interface with the HTTP API. To make the specification easy to understand the document details examples wherever possible. In the section titled "Examples" examples outlining various use cases have been provided, please go through them for a better understanding of the API. If you have downloaded the HTTP API package, it also contains a web archive (WAR) file that implements a stub for the interface. You can use this stub to produce various error messages and the success response, in single as well as multiple responses format to be able to better understand the API and to help you develop an accurate client for the API.

Since the API is implemented in Java, you would have to deploy the WAR file in Web Server/Container that serves Java Servlet Classes such as Apache Tomcat, Glassfish etc.

3. HTTP URL:

The HTTP URL for sending sms is as mentioned below:

<http://192.168.0.11:8080/HttpBox/sendsms>

4. HTTP POST Parameters:

Table 1: HTTP POST Paramaters		
Parameter	Description	Comments
username (m)	Username for account	
password (m)	Password for username	
msgtext (m)	Message text	
sender (m)	Source Id	
Destination (m)	Mobile number Min.: 6 digits Max.: 18 digits	Mobile numbers must be prefixed with country code. Multiple numbers should be comma separated not exceeding more than 50 numbers.
type (o)	Type of the message. Should be either 'text' or 'unicode' Case is insensitive.	If absent the type is assumed to be 'text'
mclass (o)	Message Class. Valid values are 0 - Direct display, 1 - Send To mobile, 2 - Send To SIM, 3 - Send To SIM Toolkit	If absent assumed to be 1.
dlr-mask (o)	Delivery Report Mask. Should be a sum of the values for which a delivery report is requested. 1 -	The final value of the dlr-mask should be a sum total of the values of all the statuses for which a report

	Delivery success, 2 – Delivery Failure, 4 – Message Buffered, 8 – SMSC Submit, 16 – SMSC Reject, 32 – SMSC Intermediate Notifications Note: The final delivery report sent back depends on the type and implementation of the SMSC. Some type of SMSCs do not support buffered messages, some SMSCs do not implement all types of delivery reports.	is needed. For e.g. If you need a report for delivery success, delivery failure and SMSC Reject. Your dlr-mask should be 1 + 2 + 16 = 19. If absent it is assumed that a delivery report is not requested for. If a wrong summation value is sent, the dlr-mask is assumed to be 1 so that only delivery success will be reported.
--	--	---

o – optional
m – mandatory

Note: All the parameters and their values should be URL UTF-8 encoded while sending through the HTTP POST method.

5. Success Response:

Table 2: Success Response	
Response	Description
OK:<dest>:<MessageID>	On a successful send, the response constitutes of the letters “OK” (without the speech marks) the destination (mobile number) the response refers to and a unique ID generated for the SMS. The literal string OK, the destination and the id are all separated by the colon (:) character. In case of multiple responses, two responses are separated from each other by a single semi-colon (;) character.
Example: OK:919812345678:2fb65852-535f-4c10-9944-5d526d81baab	

So in summary the success response takes a pattern identical to this:

OK:<dest>:<Message_ID>[;OK:<dest>:<Message_ID>]*

Note:

* - occurrence of the entire group zero or more times

6. Error Response:

An error response takes the following form:

Table 3: Error Response	
Response	Description
ERR:<Error_Code>:<Error_Msg>	The error response contains the literal string “ERR” (without the speech marks) signaling that an error has occurred, the error code representing the error occurred and a short message describing the error. The error codes and the error messages are described in Table 4 in the 'Errors' section.

Example: ERR:5001:Auth_Failed

7. Errors:

Table 4: Errors		
Error Code	Short Error Message	Description
5001	Auth_Failed	Bad Username/Password
5002	Sender_Len_Inv	Sender length should be not more than 18 digits in case of numeric sender Ids and not more than 11 in case of alphanumeric sender ids. Note: In case of India numbers (+91) alphanumeric senders should be no more than 8 characters long.
5003	Sender_Inv	If the sender is not among the list of approved senders for the user account. Note: Whether you will get this error depends on the implementation. If sender ids need to approved only for a specific SMSC, so that for other SMSCs the sender ids are open, this error wouldn't be generated here. In any case this error code is reserved for this error.
5004	Dest_Format_Inv	A destination could be between 6 to 18 digits long. Longer or Shorter destinations are considered invalid
5005	Url_Format_Err	This error occurs when any of the mandatory parameters are absent or if any of the parameter values (other than the ones specified above) are incorrectly set.
5006	Acct_Err	Errors relating to user account such as Insufficient Balance etc.
5007	Internal_Err	This error may be thrown as a response to any error occurring within the http process or the server that it is hosted on. An example of this is a Database error. Please try again after some time in case of this error.
5008	Unknown_Error	Any errors apart from the ones mentioned above may be reported through this code. Such as errors that may not be foreseen currently but may be added later on.

8. Examples:

In this section we provide examples that explain using various parameters for sending different types of SMS.

For the purpose of the examples we assume the following:

```
username = foo
password = bar
sender = test
destination = X (where X is to be replaced by the mobile number)
```

and we also assume that the user account has sufficient balance.

Ex. 1: Sending a typical text SMS (Skipping optional params)

```
http://<host-name>:<port>/sendsms?
username=foo&password=bar&msgtext=Hello+World&sender=test&destination=X
```

Description: In this example we send a typical text SMS. Since the default for Type is 'text' we can omit the parameter. Also since the default for mclass is 1, which is Send to mobile and which is what we want to do here we omit that parameter too. Finally we do not need a delivery report for this message so we skip that parameter too. Remember when the dlr-mask parameter is absent it is assumed that the delivery report is not asked for.

Ex. 2: Sending a typical text SMS (Including all params)

```
http://<host-name>:<port>/sendsms?
username=foo&password=bar&msgtext=Hello+World&sender=test&destination=X&type=
text&mclass=1&dlr-mask=0
```

Description: This example achieves the exact same thing as does Ex. 1, only this example sets all the parameters even if we are just using their default values.

Ex. 3: Sending a Unicode SMS with a request for delivery report

```
http://<host-name>:<port>/sendsms?
username=foo&password=bar&msgtext=00480065006C006C006F00200057006F0072006C006
4&sender=test&destination=X&type=unicode&mclass=1&dlr-mask=3
```

Description: Here, in this example we are sending a unicode message which we let the API know by setting the type to 'unicode' we also set the dlr-mask to 3 which means that we need a delivery report for a success as well as failure (1+2)

Ex. 4: Sending a direct display (flash) message.

```
http://<host-name>:<port>/sendsms?
username=foo&password=bar&msgtext=Hello+World&sender=test&destination=X&type=
text&mclass=0&dlr-mask=23
```

Description: In this example we set the mclass to 0 so that the message is directly displayed onto the mobile screen (flash message) instead of sending it to the message inbox. We also set the dlr-mask parameter to 23 which means we have requested for a delivery report in case of success, failure, message buffered and SMSC Reject (1+2+4+16).