

# ADVANCE MESSAGING SYSTEM

## SMS

SMS API DOCUMENT

BAB INTERNATIONAL



Advanced Messaging System Login:

<https://bms-api.bab.sa/websmpp>

# INDEX

S. No	Content	Page No.
1	Introduction	3
2	SMPP	4
3	HTTP	8
3.1	Message Submission & Scheduling	8
3.2	Status Enquiry	13
3.3	Balance Enquiry	14
3.4	Coverage Download	14
3.5	Web DLR	15
3.6	DlrReport(XLSX)	16
4	JSON	17
5	XML	19
6	Status Code	23

## Introduction

This document provides SMPP specification including connection to advance messaging systems server, bind options and specification for sending sms over SMPP. This document also provides HTTP API used to sending SMS, collecting delivery reports along with HTTP API specification.

## SMPP

The Short Message Peer-to-Peer (**SMPP**) is a protocol used by the telecommunications industry for exchanging SMS messages between Short Message Service Centers (SMSC) and External Short Messaging Entities (ESME). The protocol is a level-7 TCP/IP protocol, which allows fast delivery of SMS messages.

The connection between application and AMS SMPP Server is SMPP version 3.4

***BIND Operation:*** The purpose of the SMPP bind operation is to register an instance of an ESME with the SMSC system and request an SMPP session over this network connection for the submission or delivery of messages. Thus, the Bind operation may be viewed as a form of SMSC login request to authenticate the ESME entity wishing to establish a connection.

***BIND Mode:*** User may bind as either Tx (Transmitter), Rx (Receiver) or TRx (Transceiver).

***Tx:*** User bound as a Transmitter is authorized to send short messages and to receive the corresponding SMPP responses.

***Rx:*** User bound as a Receiver is authorized to receive short messages and to return the corresponding SMPP message responses.

***TRx:*** User bound as a Transceiver is allowed to send & receive messages over a single SMPP session.

***\*Users are allowed to bound as only 1 TRx/Rx with multiple Tx Session.***

### Requested Parameters:

Name	Description
System id	Provided System Id
Password	Provided Password
IP	46.151.210.31
Port	8080
System type	Null

### SMPP Status Codes:

Name	Value	Description
ESME_ROK	0x00000000	No Error
ESME_RINVMSGLEN	0x00000001	Invalid Message Length
ESME_RINVCMDLEN	0x00000002	Invalid Command length
ESME_RINVCMDID	0x00000003	Invalid/Unsupported Command ID
ESME_RINVBNDSTS	0x00000004	Incorrect bind status
ESME_RALYBND	0x00000005	Already bound
ESME_RINVPRFTLG	0x00000006	Invalid Priority Flag

<b>ESME_RINVREGDLVFLG</b>	0x00000007	Invalid Registered Delivery Flag
<b>ESME_RSYSERR</b>	0x00000008	System Error
<b>ESME_RINVSRCADR</b>	0x0000000A	Invalid source address
<b>ESME_RINVDSTADR</b>	0x0000000B	Invalid destination address
<b>ESME_RINVMSGID</b>	0x0000000C	Message ID is invalid
<b>ESME_RBINDFAIL</b>	0x0000000D	Bind failed
<b>ESME_RINVPASWD</b>	0x0000000E	Invalid password
<b>ESME_RINVSYSID</b>	0x0000000F	Invalid System ID
<b>ESME_RCOVERR</b>	0x00000012	No Coverage For Requested Destination
<b>ESME_RREPLACEFAIL</b>	0x00000013	Message replacement failed
<b>ESME_RMSGQFUL</b>	0x00000014	Message queue full
<b>ESME_RINVSERTYP</b>	0x00000015	Invalid service type
<b>ESME_RINVADR</b>	0x00000022	Invalid Address
<b>ESME_RPARAMRETFAIL</b>	0x00000031	Parameter Retrieve Failed
<b>ESME_RINVPARAM</b>	0x00000032	Invalid Parameter
<b>ESME_RINVNUMDESTS</b>	0x00000033	Invalid number of destinations
<b>ESME_RINVDESTFLAG</b>	0x00000040	Invalid destination flag
<b>ESME_RINVBALANCE</b>	0x0000040B	Insufficient Balance
<b>ESME_RINVSUBREP</b>	0x00000042	Invalid Submit Replace
<b>ESME_RINVESMCLASS</b>	0x00000043	Invalid ESM class

<b>ESME_RCNTSUBDL</b>	0x00000044	Cannot Submit to Distribution List
<b>ESME_RSUBMITFAIL</b>	0x00000045	Failed To Submit
<b>ESME_RINVSRCNPI</b>	0x00000048	Invalid Source Address NPI
<b>ESME_RINVSRCNPI</b>	0x00000049	Invalid Source Address NPI
<b>ESME_RINVDSTTON</b>	0x00000050	Invalid Destination Address TON
<b>ESME_RINVDSTNPI</b>	0x00000051	Invalid Destination Address NPI
<b>ESME_RINVSYSTYP</b>	0x00000053	Invalid system type
<b>ESME_RINVREPFLAG</b>	0x00000054	Invalid replace_if_present flag
<b>ESME_RINVNUMMSGS</b>	0x00000055	Invalid number of messages
<b>ESME_RTHROTTLED</b>	0x00000058	Throttling error
<b>ESME_RPROVNOTALLWD</b>	0x00000059	Provisioning Not Allowed
<b>ESME_RINVSCHED</b>	0x00000061	Invalid scheduled delivery time
<b>ESME_RINVEXPIRY</b>	0x00000062	Invalid Validity Period value
<b>ESME_RX_T_APPN</b>	0x00000064	ESME Receiver temporary error
<b>ESME_RX_P_APPN</b>	0x00000065	ESME Receiver permanent error
<b>ESME_RX_R_APPN</b>	0x00000066	ESME Receiver reject message error
<b>ESME_RQUERYFAIL</b>	0x00000067	Message query request failed
<b>ESME_RINVMDTLEN</b>	0x0000008D	Message delivery time Length Invalid
<b>ESME_RINVREGDELLEN</b>	0x00000093	Registered Delivery Length Invalid
<b>ESME_RINVREGDEL</b>	0x0000009C	Registered Delivery Invalid

<b>ESME_RUNKNOWNERR</b>	0x000000FF	Unknown error
<b>ESME_LAST_ERROR</b>	0x0000012C	Last error code

## HTTP

The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems. This is the foundation for data communication for the World Wide Web (i.e. internet)

HTTP is a generic and stateless protocol which can be used for other purposes as well using extensions of its request methods, error codes, and headers. This document provides developers with instructions for integrating SMS messaging services into various solutions using HTTP API.

This HTTP API can be used in sending SMS, collecting delivery reports along with HTTP API specification. This document also provides SMPP specification including connection to advance messaging systems server, bind options and specification for sending sms over SMPP

Users can Use Following Services via HTTP API:

- 1) Message Submission & Scheduling
- 2) Status Enquiry
- 3) Balance Enquiry
- 4) Coverage Download

### 3.1 Message Submission & Scheduling

HTTP API supports both GET and POST Methods. Using GET Method limit of up to 100 numbers is allowed in a single request. For POST Method limit of up to 10000 numbers is allowed in a single request.

#### Description:

URL for Message Submission:



**https://bms-api.bab.sa/websmpp/websms?  
user=xxxx&pass=xx&sid=xxxxx&mno=966xxxxx&type=x&text=xxx**

or

**https://bms-api.bab.sa/websmpp/websms? accesskey=xxxxxxxxxxxxxxxx  
&sid=xxxxx&mno=966xxxxx&type=x&text=xxx**

If Response Required in JSON Format:

**https://bms-api.bab.sa/websmpp/websms?  
user=xxxx&pass=xx&sid=xxxxx&mno=966xxxxx&type=x&text=xxx&respformat=j  
son**

URL for Message Scheduling:

**https://bms-api.bab.sa/websmpp/websms?  
user=xxxx&pass=xx&sid=xxxxx&mno=966xxxxx&type=x&text=xxx  
&gmt=xxxxx&sctime=xxxxxxxxxxxx**

### **Using configured group of address book:**

**https://bms-api.bab.sa/websmpp/websms/groupsms?  
user=xxxx&pass=xx&sid=xxxxx&group=GR1&limit=100&type=x&text=xxx**

### **Requested Parameters (mandatory):**

Name	Description
<b>user</b>	Provided System Id
<b>pass</b>	Provided Password
<b>accesskey</b>	Can be used instead of user & pass

<b>sid</b>		Source Number (Numeric/Alphanumeric)
<b>mno/group</b>		Destination Number/Group or Comma separated Destination Numbers (Including Country Code)/Groups
<b>type</b>		Message Type (1,2,3,4)
<b>text</b>		Message Content

**Optional Parameters (Scheduling):**

Name	Description
<b>gmt</b>	GMT i.e. +0530
<b>schttime</b>	Schedule time (yyyyMMddhhmm)
<b>limit</b>	Destination number limit per group

**Optional Parameters (others):**

Name	value	Description
<b>respformat</b>	<b>json</b>	If response required in json format

**Message Type & Length:**

Value	Message Type	Length (1 SMS)
-------	--------------	----------------

1	English	160
2	Unicode	280
3	Special Character	160
4	Arabic	70

**Examples:**

- **English Message:**

```
https://bms-api.bab.sa/websmpp/websms?  
user=testUser&pass=12345&sid=Test&mno=966xxxxx&type=1  
&text=testing English message
```

- **Unicode Message:**

```
https://bms-api.bab.sa/websmpp/websms?  
user=testUser&pass=12345&sid=Test&mno=966xxxxx&type=2  
&text= 00740065007300740069006E006700200075006E006  
90063006F006400650020006D006500730073006100670065
```

- **Special Character Message:**

```
https://bms-api.bab.sa/websmpp/websms?  
user=testUser&pass=12345&sid=Test&mno=966xxxxx&type=3  
&text=testing $@ Character
```

- **Arabic:**

[https://bms-api.bab.sa/websmpp/websms?  
user=testUser&pass=12345&sid=Test&mno=966xxxxx&type=4 &text=  
عربية رسالة اختبار](https://bms-api.bab.sa/websmpp/websms?user=testUser&pass=12345&sid=Test&mno=966xxxxx&type=4 &text=عربية رسالة اختبار)

- **Scheduling:**

[https://bms-api.bab.sa/websmpp/websms?  
user=testUser&pass=12345&sid=Test&mno=966xxxxx&type=1  
&text=testing Schedule &gmt=+0530&schtime=201605101422](https://bms-api.bab.sa/websmpp/websms?user=testUser&pass=12345&sid=Test&mno=966xxxxx&type=1&text=testing Schedule &gmt=+0530&schtime=201605101422)

**Note: The characters [ ] { } | \ ~ ^ will be counted twice. The characters & and # are restricted in message content.**

## **Response Format**

### **On Successful Submission (Default Format):**

Response: 1507211343399277335

### **On Successful Submission (JSON Format):**

```
{"Response":["1912181513205684021"]}
```

### **On Failed (Possible Errors):**

ERROR - HTTP01 --> Invalid HTTP URL Format

ERROR - HTTP02 --> Invalid query string

ERROR - HTTP03 --> Message Submission Failed

ERROR - HTTP04 --> Invalid Password

ERROR - HTTP05 --> Invalid System ID

ERROR - HTTP06 --> Invalid Sender ID

ERROR - HTTP07 --> Invalid Type

ERROR - HTTP08 --> Invalid Message ID

ERROR - HTTP09 --> Invalid Destination Address

ERROR - HTTP10 --> Invalid number of messages

ERROR - HTTP11 --> Requested service is unavailable

ERROR - HTTP12 --> Submit Failed

ERROR - HTTP13 --> Query request failed

ERROR - HTTP14 --> Bind failed

ERROR - HTTP15 --> Not Authorized for the request.

ERROR - HTTP16 --> Invalid Request

ERROR - HTTP17 --> Invalid Message Length

ERROR - HTTP18 --> INSUFFICIENT CREDITS/BALANCE

ERROR - HTTP19 --> Invalid DATA CODING SCHEME

ERROR - HTTP20 --> Invalid ESM

ERROR - HTTP21 --> Invalid HEADER

ERROR - HTTP22 --> Invalid Schedule Time

ERROR - HTTP23 --> Routing Error

ERROR - HTTP24 --> Exceeded number of Destination

ERROR - HTTP25 --> No Record Found

ERROR - HTTP26 --> Account Expired

## 3.2 Status Enquiry

After Successful Submission, User can get Message Status via Following URL:

**<https://bms-api.bab.sa/websmpp/websmsstatus?respid=2541524>**

Resulting status would be one of them from the following table:

Name		Description
<b>ATES</b>		Pending
<b>DELIVRD</b>		Delivered
<b>UNDELIV</b>		Undelivered
<b>EXPIRED</b>		Message Validity Period Expired
<b>REJECTD</b>		Rejected
<b>ACCEPTD</b>		Accepted
<b>DELETED</b>		Deleted
<b>UNKNOWN</b>		Unknown

### 3.3 Balance Enquiry

After Successful Submission, User can get Message Status via Following URL:

**<https://bms-api.bab.sa/websmpp/balanceReport?userid=systemid&password=pwd>**

or

<https://bms-api.bab.sa/websmpp/balanceReport?accesskey=xxxxxxxxxxxxxxxx>

- The parameters userid & password are mandatory. Failing any would result in a negative response.

### 3.4 Coverage Download

User can get Coverage via Following URL:

<https://bms-api.bab.sa/websmpp/downloadcoverage?userid=systemid&password=pwd&format=x>

or

<https://bms-api.bab.sa/websmpp/downloadcoverage?accesskey=xxxxxxxxxxxxxxxx&format=x>

*Format is optional parameter. If not specified the default format will be 1.*

#### Requested Formats:

Value	Downloaded Format
1	XLS
2	PDF
3	CSV

### 3.5 Web DLR

User can get DLR via Provided Web Link. The URL for each client would look like '<http://www.test.com/dlr/dlr.aspx>'. The method used for posting delivery report onto the client's URL is POST. After receiving DLR, the link must return **Ok** as acknowledgement.

The parameters appended to the URL would be as below:

Parameter	Discription
<b>MessageId</b>	Message Id provided on submission
<b>Source</b>	Source Address
<b>Destination</b>	Destination Number
<b>Status</b>	Status of the Request (i.e. DELIVRD,UNDELIV,ATES)
<b>SubmitDate</b>	Time of Submission (yyMMddHHmm)
<b>DoneDate</b>	Time of Delivery Report Receiving( yyMMddHHmm )
<b>BatchId</b>	Batch Id As Immediate Response (for XML & JSON)

### 3.6 DlrRepot(XLSX)

DLR Report in XLSX format can be downloaded via below Web Link:

**[https://bms-api.bab.sa/websmpp/api/dlrreport?username=xxxx&pass=xxxx&from\\_date=20210821124545&to\\_date=20210821164545&is\\_content=1&sender=XYZ](https://bms-api.bab.sa/websmpp/api/dlrreport?username=xxxx&pass=xxxx&from_date=20210821124545&to_date=20210821164545&is_content=1&sender=XYZ)**

or **[https://bms-](https://bms-api.bab.sa/websmpp/api/dlrreport?accesskey=xxxxxxxxxxxxxxxx&from_date=20210821124545&to_date=20210821164545&is_content=1&sender=XYZ)**

**[api.bab.sa/websmpp/api/dlrreport?accesskey=xxxxxxxxxxxxxxxx  
&from\\_date=20210821124545&to\\_date=20210821164545&is\\_content=1&sender=XYZ](https://bms-api.bab.sa/websmpp/api/dlrreport?accesskey=xxxxxxxxxxxxxxxx&from_date=20210821124545&to_date=20210821164545&is_content=1&sender=XYZ)**

### **Requested Parameters (mandatory):**



<b>username</b>	Provided System Id
<b>pass</b>	Provided Password
<b>from_date</b>	Start Date (yyyyMMddHHmmss)
<b>to_date</b>	End Date (yyyyMMddHHmmss)

### Requested Parameters (Optional):

<b>sender</b>	Source Address(Numeric/Alphanumeric)
<b>Is_content</b>	Default value is 0.Values can be 0 or 1. If content is needed value must be 1

## JSON

Users can Use Following Services via JSON API:

1. Message Submission
2. Status Enquiry

**4.1 Message Submission:** The URL used to post XML formatted data is:

**<https://bms-api.bab.sa/websmpp/json/sms>**

Request Format Example[1]:

```
{"campaign":{"username":"testUser","password":"xx",  
"format":"1","sender":"JSON-1",  
"gsm":["919826321321","919589321322"], "text":"testing normal json  
api","type":"1"}}
```

or

```
{"campaign":{"accesskey":"xxxxxxxxxxxxxxxx",  
"format":"1","sender":"JSON-1",  
"gsm":["919826321321","919589321322"], "text":"testing normal json  
api","type":"1"}}
```

*or (using configured group of address book)*

```
{"campaign":{"username":"testUser","password":"xx",  
"format":"1","sender":"JSON-1",  
"group":["GR1","GR2"], limit:"100","text":"testing normal json api","type":"1"}}
```

Request Format Example[2]:

```
{"campaign":{"username":"testUser","password":"xx","format":"2",  
"type":"1","sender":"JSON-2", "custom":[{"gsm":"919826321321","text":"testing  
message 1"}, {"gsm":"919589321322","text":"testing message 2"}]}}
```

**Response:** After Submission User will get Response as below:

```
{"response":{"batch_id":"0302201404163965268","status":"000"}}
```

batch\_id is unique id and used to identify all the messages( on all gsm numbers) associated with this request.

Status will be 000 on successful Submission. For Other status please check the status Code table.

**4.2 Status Enquiry:** After Successful Submission, User can get Message Status via Following URL:

**<https://bms-api.bab.sa/websmpp/json/status>**

To Enquire the Status of Submissions, A specified format is Required as below:

```
{"enquiry":{"password":"xx","batch_id":"0302201420337911363","username":"testUser"}}
```

or

```
{"enquiry":{"accesskey":"xxxxxxxxxxxxxxxx","batch_id":"0302201420337911363"}}
```

**Resulted Format :** The Resulted JSON Format Will be as below:

```
{"result":{"batch_id":"0302201420337911363","report":[{"request_status":"000","dlr_status":"UNDELIV","messageid":"2002031420338711051","msisdn":"919826321321"},{"request_status":"000","dlr_status":"DELIVRD","messageid":"2002031420338811061","msisdn":"919589321322"}]}}
```

## XML

Users can Use Following Services via XML API:

1. Message Submission
2. Status Enquiry

**5.1 Message Submission:** The URL used to post XML formatted data is:

**<https://bms-api.bab.sa/websmpp/xmlsms>**

Request Format Example[1]:

```
<?xml version="1.0" encoding="UTF-8"?>
<campaign format="1">
<SMS>
<username>testUser</username>
<password>pass</password>
<type>1</type>
<sender>TEST-XML1</sender>
<text>testing xml request via http</text>
<recipients>
<gsm>966231213</gsm>
<gsm>966321987</gsm>
</recipients>
</SMS>
</campaign>
```

or

```
<?xml version="1.0" encoding="UTF-8"?>
<campaign format="1">
<SMS>
<accesskey>xxxxxxxxxxxxxxxxxx</accesskey>
<type>1</type>
<sender>TEST-XML1</sender>
<text>testing xml request via http</text>
<recipients>
<gsm>966231213</gsm>
<gsm>966321987</gsm>
</recipients>
</SMS>
</campaign>
```

or (**using configured group of address book**)

```
<?xml version="1.0" encoding="UTF-8"?>
<campaign format="1">
<SMS>
<username>testUser</username>
<password>pass</password>
<type>1</type>
<sender>TEST-XML1</sender>
<text>testing xml request via http</text>
<recipients>
<group>GR1</group>
<group>GR2</group>
</recipients>
<limit>100</limit>
</SMS>
</campaign>
```

### Request Format Example[2]:

```
<?xml version="1.0" encoding="UTF-8"?>
<campaign>
<SMS format="2">
<username>testUser</username>
<password>pass</password>
<type>1</type>
<sender>TEST-XML1</sender>
<custom>
<gsm>966231211</gsm>
<text>testing xml custom request 1 via http</text>
</custom>
<custom>
<gsm>966231212</gsm>
<text>testing xml custom request 2 via http</text>
</custom>
```

```
<custom>  
<gsm>966231213</gsm>  
<text>testing xml custom request 3 via http</text>  
</custom>  
</SMS>  
</campaign>
```

**Response:** After Submission User will get Response as below:

```
<?xml version="1.0" encoding="UTF-8"?>  
<response>  
<status>000</status>  
<batch_id>171117172801751</batch_id>  
</response>
```

batch\_id is unique id and used to identify all the messages( on all gsm numbers) associated with this request.

Status will be 000 on successful Submission. For Other status please check the xml status Code table.

**5.2 Status Enquiry:** After Successful Submission, User can get Message Status via Following URL:

**<https://bms-api.bab.sa/websmpp/xmlstatus>**

To Enquire the Status of Submissions, A specified format is Required as below:

```
<?xml version="1.0" encoding="UTF-8"?>  
<enquiry>  
<username>testUser</username>  
<password>pass</password>  
<batch_id>171117104025321</batch_id>  
</enquiry>
```

or

```
<?xml version="1.0" encoding="UTF-8"?>
<enquiry>
<accesskey>xxxxxxxxxxxxxxxx</accesskey>
<batch_id>171117104025321</batch_id>
</enquiry>
```

**Resulted Format :** The Resulted XML Format Will be as below:

```
<?xml version="1.0" encoding="UTF-8"?>
<result>
<batch_id>171117172801751</batch_id>
<report>
<messageid>521452541</messageid>
<msisdn>966231211</msisdn>
<dlrstatus>DELIVRD</dlrstatus>
<status>000</status>
</report>
<report>
<messageid></messageid>
<msisdn>966231212</msisdn>
<dlrstatus></dlrstatus>
<status>006</status>
</report>
</result>
```

## Status Code Table

This section Describes Status code returned while using api

Code	Name	Description
000	NO_ERROR	success
001	INVALID_LOGIN	Either System_id or password is invalid
002	INVALID_REQUEST	Required parameters are missing or Invalid parameters or invalid values of parameters
003	ACCOUNT_EXPIRED	Account has been Expired
004	INSUF_BALANCE	Insufficient balance
005	INVALID_DEST_ADDR	Invalid Destination Address
006	NO_COVERAGE	Requested Destination is not Allowed
007	SYSTEM_ERROR	System Error
008	INVALID_SENDER	Invalid Source Address
009	INVALID_TEXT	Invalid message content
010	INVALID_TYPE	Invalid Message Type
050	ACCESS_DENIED	Either Web Access Not Allowed or Account is Blocked
051	IOSTREAM_ERROR	Input Stream Error
052	INVALID_BATCHID	Invalid Batch Id (must be 15 digits)
053	NO_RECORDS	Records are not available for provided Batch Id
060	INVALID_AMOUNT	Invalid Amount For Recharge
061	INVALID_OPERATION	Invalid operation (must be <b>plus</b> or <b>minus</b> )
062	INVALID_BALANCE_	Invalid mode (must be <b>credit</b> or



	MODE	wallet)
063	INVALID_TARGET_USER	Invalid target User to recharge
064	INVALID_API_ACCESS_IP	Access IP not valid

**Note:**

For more information don't hesitate to contact technical support:

Email: [noc-ksa@bab.sa](mailto:noc-ksa@bab.sa)

Mobile: 0597199990