HTTP API FOR SMS SERVER

1. SEND-RECEIVE SMS

Using this API, The Server can act as Origin and send sms to its own copy,or act as termination and receive sms.

Receive SM Request API:

Parameters:

Parameter	Value	Required
apikey	apikey provided by termination	√
secretkey	secretkey provided by termination	√
Content	JSON formatted array of sms. Described bellow	√

Request endpoint:

a. To send data as JSON object (supports multiple sms with single request)

http://192.168.18.119:8585/send

b. To send data through simple parameters

http://192.168.18.119:8585/sendtext

Sample request:

1. without hash request

a. *http://192.168.18.119:8585/send?apikey=5d2a&secretkey=465&content=[{"callerID":"8801847","toUser":"452621,568468","messageContent":"abcd"},{"callerID":"9101847","toUser":"912621,918468","messageContent":"abcd"}]*

b. *http://192.168.18.119:8585/sendtext?apikey=5d2a&secretkey=465&callerID=8801847&toUser=452621&messageContent=abcd*

2.with hash request

a. *http://192.168.18.119:8585/send?apikey=5d2a&secretkey=465&content=[{"callerID":"8801847","toUser":"452621,568468","messageContent":"abcd"},{"callerID":"9101847","toUser":"912621,918468","messageContent":"abcd"}]&hash=dsfasdasfsdf*

b. *http://192.168.18.119:8585/sendtext?apikey=5d2a&secretkey=465&callerID=8801847&toUser=452621&messageContent=abcd&hash=asdfsadds*

content parameter explained:

There are 3 parameters for every content

1. callerID – Senders cell/identifying number
2. toUser – Semicolon separated receivers numbers.
3. messageContent – SMS content.

For example –

a. *&callerID=8801847&toUser=452621&messageContent=abcd*
b. data structured as json object for supporting multi sms with one request

```
[{
"callerID":"8801847",
"toUser":"452621;568468",
"messageContent":"abcdef"
},
{
"callerID":"9101847",
"toUser":"912621;918468",
"messageContent":"Hi"
}]
```

Response:

Example:

```
{"Status":"0","Text":"ACCEPTD","Message_ID":"444"}
```

API Response Status:

Case	Response
If user not provided/Deleted	{“Status”:”109”,”Text”:”REJECTD”,"Message_ID":""}
If wrong password/ not provided	{“Status”:”108”,”Text”:”REJECTD”,"Message_ID":""}
If content not provided	{“Status”:”114”,”Text”:”REJECTD”,"Message_ID":""}
if internal server error occurs	{“Status”:”101”,”Text”:”REJECTD”,"Message_ID":""}
If request successful	{“Status”:”0”,”Text”:”Acceptd”,"Message_ID":"message_id"}
If request failed	{“Status”:”1”,”Text”:”REJECTD”,"Message_ID":""}
If TPS limit exceed	{“Status”:”137”,”Text”:”REJECTD”,"Message_ID":""}

Send SM Request:

While sending sms from this server, the URL should contain two parts, the “request” part **(base URL)** should look like this

a. for json structure

http://192.168.18.119:8585/send?apikey=5d2a&secretkey=465

and the “query” part **(query URL)** look like this

&content=[{"callerID":"REPLACE_FROM","toUser":"REPLACE_TO","messageContent":"REPLACE_MESSAGE"}]

b. for simple request

http://192.168.18.119:8585/sendtext?apikey=5d2a&secretkey=465

and the “query” part **(query URL)** look like this

&callerID=REPLACE_FROM&toUser=REPLACE_TO&messageContent=REPLACE_MESSAGE

where

Key	meaning
REPLACE_FROM	callers ID
REPLACE_TO	receivers ID
REPLACE_MESSAGE	message text

Parsing Response from Termination

Response:

Termination will send response after receiving an sms.Origination have to save response pattern for each termination as “Response Text”. The format of the response after receiving sms should be explained in the “submit” part of the “Response Text” field of an http terminating client.This data have to be provided while adding the terminating client at the billing section of the origination.

```
{

        "submit": {

                "format": "json",

                "status": "Text",

                "successStatus": "ACCEPTD",

                "deliveryStatus": "DELIVRD",

                "messageId": "Message_ID",

                "fieldSeparator": "",

                "pairSeparator": ""


        },

        "deliver": {

                "format": "json",

                "status": "Text",

                "successStatus": "DELIVRD",

                "failedStatus": ["REJECTD"],

                "messageId": "Message_ID",

                "fieldSeparator": "",

                "pairSeparator": ""

        }

}
```

Explanation of Response Text:

Key	Meaning
format	format of the response
status	keyword in the response which contains the status text
successStatus	status text for accepted sms
deliveryStatus	status text for accepted sms in case the terminating provider doesn’t provide any delivery data. in this case ‘successStatus’ field should be blank.

messageId	the id provided by this termination..
fieldSeparator	character to separate parameters in case of the response in 'Text' format
pairSeparator	character to separate key-value pairs in case of the response in 'Text' format

Example:

1. If the response of a termination is in json format and the response is like this

   {"Status":"0","Text":"ACCEPTD","Message_ID":"444"}

   then the "Response Text" for that termination should look like this

   { "submit": { "format": "json", "status": "Text", "successStatus": "ACCEPTD", "deliveryStatus": "", "messageId": "Message_ID", "fieldSeparator": "", "pairSeparator": "" }, "deliver": { "format": "json","status":"Text","successStatus":"DELIVRD","failedStatus":["REJECTD"],"messageId":"Message_ID", "fieldSeparator": "", "pairSeparator": "" } }

2. If the response of a termination is in json format and the response is like this

   {"Status":"0","Text":"DELIVRD","Message_ID":"444"}

   then the "Response Text" for that termination should look like this

   { "submit": { "format": "json", "status": "Text", "successStatus": "" , "deliveryStatus": "DELIVRD", "messageId": "Message_ID", "fieldSeparator": "", "pairSeparator": "" }, "deliver": { "format": "json","status":"Text","successStatus":"DELIVRD","failedStatus":["REJECTD"],"messageId":"Message_ID", "fieldSeparator": "", "pairSeparator": "" } }

3.

sample API Response:

```
{
        "Status":"0",
        "Text":"ACCEPTD",
        "Message_ID":"777"
}
```

Next two APIs are for asking/providing SMS Status after sending/receiving SMS. A termination can provide the origin with "delivery URL", then it's up to the origination to ask for status. or the Origination can provide the termination with “delivery URL”, then it’s up to the termination to inform the origination the sms status. This server implements both of this features So that it’s possible to connect with any of these servers.

2. *ASK SMS STATUS*

If the Termination provides “delivery URL” ,then it’s up to the origination to do query on the termination for status receipt after sending the sms.

Using this API Origin asks SMS delivery receipt from this server. This “delivery URL” have to be added while creating Terminating HTTP client. server must provide the origination with apikey and secretkey for validating the query.

Query the server for status

Parameters:

Parameter	Value	Required
apikey	apikey provided by termination	√
secretkey	secretkey provided by termination	√
messageid	id provided by the termination	√

Request format for this server:

1. for single message ID

http://192.168.18.119:8585/getstatus?apikey=5d2a5cebf00d66bb65f30b5a81c882e4&secretkey=46586ec2&messageid=REPLACE_ID

2. for multiple message ID

http://192.168.18.119:8585/getmultistatus?apikey=5d2a5cebf00d66bb65f30b5a81c882e4&secretkey=46586ec2&messageids=REPLACE_ID

example:

1. for single message ID

http://192.168.18.119:8585/getstatus?apikey=5d2a5cebf00d66bb65f30b5a81c882e4&secretkey=46586ec2&messageid=7777

2. for multiple message ID

http://192.168.18.119:8585/getmultistatus?apikey=5d2a5cebf00d66bb65f30b5a81c882e4&secretkey=46586ec2&messageids=7777,7778,7779

Response:

Example:

1. for single message ID

```
{"Status":"0","Text":"DELIVRD","Message_ID":"7777","Delivery Time":"1581330747547"}
```

2. for multiple message ID

```
[{"Status":"0","Text":"DELIVRD","Message_ID":"7777","Delivery
Time":"1581330747547"},{"Status":"0","Text":"DELIVRD","Message_ID":"7778","Delivery
Time":"1581330747547"},{"Status":"0","Text":"DELIVRD","Message_ID":"7779","Delivery
Time":"1581330747547"}]
```

API Response Status:

Case	Response
If api key isn’t provided	{“Status”:”109”,”Text”:”REJECTD”}
If wrong password/ not provided	{“Status”:”108”,”Text”:”REJECTD”}
If message id isn’t provided	{“Status”:”114”,”Text”:”REJECTD”}
If message id isn’t valid or query has been done already for the same id	{“Status”:”114”,”Text”:”REJECTD”}
if internal server error occurs	{“Status”:”101”,”Text”:”REJECTD”}
if Authorization failed	{“Status”:”-42”,”Text”:”REJECTD”}
If request failed	{“Status”:”1”,”Text”:”REJECTD”}
If request pending	{“Status”:”2”,”Text”:”PENDING”}
If request sent	{“Status”:”4”,”Text”:”SENT”}
If sms request successful	{“Status”:”0”,”Text”:”DELIVRD”}

Response as Termination:

Response:

Termination will send a response after receiving an query.Origination has to save response pattern for each termination as “Response Text”. The format of the response after receiving sms should be explained in the “deliver” part of the “Response Text” field of an http terminating client.This data have to be provided while adding the terminating client at the billing section of the origination.

response format should be explained in “Response Text” fields “deliver” part in http terminating client

```
{
        "submit": {
                "format": "json",
                "status": "Text",
                "successStatus": "ACCEPTD",
                "deliveryStatus": "DELIVRD",
                "messageId": "Message_ID",
                "fieldSeparator": "",
                "pairSeparator": ""

        },
        "deliver": {
                "format": "json",
                "status": "Text",
                "successStatus": "DELIVRD",
                "failedStatus": ["REJECTD"],
                "messageId": "Message_ID",
                "fieldSeparator": "",
                "pairSeparator": ""

        }
}
```

Explanation of Response Text:

Key	Meaning
format	format of the response
status	keyword in the response which contains the status text

successStatus	status text for accepted sms
failedStatus	array of status text for rejected sms
messageId	the id provided by this termination
fieldSeparator	character to separate parameters in case of the response in ‘Text’ format
pairSeparator	character to separate key-value pairs in case of the response in ‘Text’ format

Example:

If the response of a termination is in json format and the response is like this

```
{"Status":"0","Text":"DELIVRD","Message_ID":"444"}
```

then the “Response Text” for that termination should look like this

```
{"submit":{"format":"json","status":"Text","successStatus":"ACCEPTD","deliveryStatus": "", "messageId": "Message_ID","fieldSeparator": "","pairSeparator": "" }, "deliver": { "format": "json", "status": "Text", "successStatus": "DELIVRD", "failedStatus": ["REJECTD"], "messageId": "Message_ID","fieldSeparator": "","pairSeparator": "" } }
```

sample API Response:

```
{
        "Status":"0",
        "Text":"DELIVRD",
        "Message_ID":"777"
}
```

3.SEND SMS STATUS

If the Origination provides "delivery URL" ,then it's up to the Termination to inform the Origination the status receipt after receiving, processing and sending the sms to the targeted clients.

Using this API server sends SMS delivery receipt to the Origination. This "delivery URL" has to be provided while creating Originating HTTP clients in the billing section for Origination .

Sending status to the origin

1. using GET Method

delivery url format of origination as set in web:

http://abc.com/submitstatus?apikey=5d2a5cebf00d66bb65f30b5a81c882e4&secretkey=46586ec2&status=REPLACE_STATUS&text=REPLACE_TEXT&messageid=REPLACE_ID

Valid delivery url will be created by replacing the following parts of the given url with appropriate data -

Key	meaning
REPLACE_STATUS	status code
REPLACE_TEXT	status text
REPLACE_ID	message id

Formation or Response Text in Origin:

then the "Response Text" for that termination should look like this

```
{"submit":{"format":"json","status":"Text","successStatus":"ACCEPTD","deliveryStatus": "", "messageId": "Message_ID","fieldSeparator":  "","pairSeparator":  ""  },  "deliver":  {  "format":  "",  "status":  "text", "successStatus":  "DELIVRD",  "failedStatus":  ["REJECTD"],  "messageId":  "message","fieldSeparator": "","pairSeparator": "" } }
```

2. using POST Method

delivery url format of origination as set in web:

http://abc.com/submitstatus?apikey=5d2a5cebf00d66bb65f30b5a81c882e4&secretkey=46586ec2

parameters= Status:value, Message_ID:value, Text:value

example: Status:0, Message_ID:775000, Text:DELIVRD

Formation or Response Text in Origin:

if the termination submits the values as post parameters, then the "Response Text" for that termination should look like this

{"submit":{"format":"json","status":"Text","successStatus":"ACCEPTD","deliveryStatus": "", "messageId": "Message_ID","fieldSeparator": "","pairSeparator": "" }, "deliver": { "format": "", "status": "Text", "successStatus": "DELIVRD", "failedStatus": ["REJECTD"], "messageId": "Message_ID","fieldSeparator": "","pairSeparator": ""}}

Valid delivery data will contain following types of data -

Status	Key in GET Method	meaning
Successfully delivered	REPLACE_STATUS	0
	REPLACE_TEXT	DELIVRD
	REPLACE_ID	message id
Failed to deliver	REPLACE_STATUS	1
	REPLACE_TEXT	REJECTD
	REPLACE_ID	message id
Authorization Failed	REPLACE_STATUS	-42
	REPLACE_TEXT	REJECTD
	REPLACE_ID	message id

Response as origin from server

Response Example:

{"Status":"0","Text":"SUCCESS","Message_ID":"444"}

API Response Status:

Case	Response

If user not provided/Deleted	{“Status”:”109”,”Text”:”REJECTD”}
If wrong password/ not provided	{“Status”:”108”,”Text”:”REJECTD”}
If content not provided	{“Status”:”114”,”Text”:”REJECTD”}
If status doesn’t match with configured status	{“Status”:”137”,”Text”:”REJECTD”}
if internal server error occurs	{“Status”:”101”,”Text”:”REJECTD”}
If request successful	{“Status”:”0”,”Text”:”SUCCESS”}
If request failed	{“Status”:”1”,”Text”:”REJECTD”}