

# KARIX PowerAPI – SMS – Query String

## User Guide

Date: 06/04/2018

Version no.: 1.1

# TABLE OF CONTENTS

<b>1</b>	<b>PREFACE .....</b>	<b>4</b>
<b>2</b>	<b>A FEW ASSUMPTIONS.....</b>	<b>4</b>
<b>3</b>	<b>SAMPLE HTTP URL FORMAT .....</b>	<b>4</b>
<b>4</b>	<b>REQUIRED ENVIRONMENT .....</b>	<b>5</b>
<b>5</b>	<b>KARIX API PARAMETERS SPECIFICATION .....</b>	<b>5</b>
<b>6</b>	<b>SUCCESSFUL TRANSMISSION OF SMS .....</b>	<b>7</b>
<b>7</b>	<b>ERROR CODES .....</b>	<b>7</b>
<b>8</b>	<b>SENDING NORMAL TEXT MESSAGE (160 CHARACTERS) .....</b>	<b>7</b>
<b>9</b>	<b>SENDING TEXT MESSAGE OF MORE THAN 160 CHARACTERS.....</b>	<b>8</b>
<b>10</b>	<b>SENDING UNICODE CHARACTER .....</b>	<b>8</b>
<b>11</b>	<b>SENDING BINARY MESSAGE (160 CHARACTERS) .....</b>	<b>11</b>
<b>12</b>	<b>SENDING BINARY MESSAGE OF MORE THAN 160 CHARACTERS .....</b>	<b>12</b>
<b>13</b>	<b>SENDING WAP MESSAGES .....</b>	<b>12</b>
<b>14</b>	<b>SENDING INTERNATIONAL MESSAGES.....</b>	<b>13</b>
<b>15</b>	<b>CALL BACK DELIVERY REPORT.....</b>	<b>14</b>
<b>16</b>	<b>SAMPLE URL FORMAT .....</b>	<b>15</b>
<b>17</b>	<b>POLLING DELIVERY REPORTS: .....</b>	<b>15</b>
<b>17.1</b>	<b>Key Notes for polling DLR.....</b>	<b>16</b>
<b>17.2</b>	<b>Sample response on HTTP for tilde separated: .....</b>	<b>17</b>
<b>17.3</b>	<b>Sample response on HTTP XML response:.....</b>	<b>17</b>

**18 STATUS CODES AND REASON RECEIVED ON THE URL ..... 18**

# 1 PREFACE

This guide provides specifications of the HTTP/HTTPS based API provided by Karix for the automated sending of SMS via the Internet. This guide is intended for developers planning on integrating their systems with Karix's SMS service. It also provides the information about sending the Unicode, Binary and WAP messages.

## 2 A FEW ASSUMPTIONS

This guide assumes that you are familiar with and have experience applying the following technologies and concepts:

- HTTP/HTTPS communications using the GET and POST methods for parameter passing through API.
- A programming language such as Java, C/C++, Visual Basic or PHP to integrate the HTTP URL with the Application.
- Application designed in any of the above Language should be capable of capturing the Message ID (MID).

## 3 SAMPLE HTTP URL FORMAT

The below HTTP API can be used by Client to send the messages to Karix's Server. A sample of the URL could be in the below format

### *Sample Format:*

```
http://api.instaalerts.zone/SendSMS/sendmsg.php?uname=XXX&pass=XXX  
&send=91XXXXXXXXXX&dest=91XXXXXXXXXX&msg=XXXX
```

### *Sample API Call made by Client's Application:*

```
http://api.instaalerts.zone/SendSMS/sendmsg.php?uname=Username&pas  
s=password&send=919980524000&dest=919243201476&msg=Testing
```

## 4 REQUIRED ENVIRONMENT

In order to use the HTTP/HTTPS based API, you will need to have a system that has internet connectivity on outbound port 80 (HTTP) or 443 (HTTPS). This connectivity may be direct to the internet, or may be via a firewall or proxy server.

Additionally, should you wish to receive delivery reports (DLRs) from the Karix system, you must have a web server that is able to receive incoming requests from the Internet on either port 80 (HTTP) or 443 (HTTPS). This web server does not necessarily have to be running on the same machine that is sending messages to Karix.

This is one of the simpler server-based forms of communication. It can be used either in the form of a HTTP POST or HTTP GET. We recommend POST for larger data transfer and data security. All calls to the API must be URL-encoded. The parameter names are case sensitive.

## 5 KARIX API PARAMETERS SPECIFICATION

Whenever the Client Application hits the HTTP URL, it should include the following parameters as per the requirement. The following parameters should be used in the same order as given below to call HTTP API using POST or GET method.

S.No	Parameter	Type	Description
1	uname*	String	Username Assigned to the account
2	Pass*	string	Password Assigned to the account
3	send**	string	SenderID-Source Address for the message
4	dest*	string	Destination Address/Mobile Number of the message
5	msg*	string	Message-Text Content of the message (max length should not cross 459 characters (if concatenation is on) or 160 character for 8 bit message; 280 characters for binary and 70 for Unicode)
6	prty	number	Priority-The SMSC will process it based on the priority value. If this value is not set in the URL then our application will take the default value set at the account level.

7	vp	number	Validity Period-Validity period to expire the message in the set time which can be set from 1 to 180 minutes.
8	concat	number	Concatenation-Enables sending multipart messages to the mobile.
9	intl	number	International-Enable sending the International Messages
10	udhi*	number	User Data Header-Indicating that message payload is binary message
11	dcs*	number	Data Coding Scheme-Indicate the DCS value for the message. The 0 value indicates that the message payload is 8 bit data
12	port	number	Port-Port number to send the port specific messages
13	wapurl*	string	WAP URL-Set the value for wap url
14	waptxt*	string	WAP TEXT-Set the value for WAP Text
15	unicode	number	Unicode Text- Use this value if the submitted text is Unicode (even English)
16	tag1		
17	tag2		
18	tag3		
19	tag4		
20	tag5		

**Note:**

**\* represents mandatory parameter**

**\*\* This field become mandatory only for dynamic and multiple sender id future. For static this is not a mandatory field**

## 6 SUCCESSFUL TRANSMISSION OF SMS

For each successful submission, the API would return a unique message ID (MID) for that transaction. The Client's Application should capture the MID and if a Delivery Report (DLR) can be fetched against that MID. This message ID then can be used to track the status of the message and receive the DLR against this MID through DLR URL.

## 7 ERROR CODES

The following error can be displayed while hitting the HTTP API if there is any wrong parameter entered or issue with the Account.

S.No	Error Code	Descriptions
1	0x200	Invalid Username or Password
2	0x201	Account Suspended due to some reason.
3	0x202	Invalid Source Address/Sender Id. As per GSM standard the sender ID should be within 11 characters.
4	0x203	Message Length Exceeded (more than 160 chars) if concat is set to 0
5	0x204	Message Length Exceeded (more than 459 chars) if concat is set to 1
6	0x205	DLR URL is not set
7	0x206	Only the subscribed service type can be accessed so make sure that the service type you are trying to connect with.
8	0x207	Invalid Source IP. Kindly check if the IP is responding.
9	0x208	Account Deactivated/Expired.
10	0x209	Invalid Message Length (less than 160 chars) if concat is set to 1
11	0x210	Invalid Parameter values
12	0x211	Invalid Message Length (more than 280 chars)
13	0x212	Invalid Message Length
14	0x213	Invalid Destination number

## 8 SENDING NORMAL TEXT MESSAGE (160 CHARACTERS)

While sending normal 160 character messages Client needs to ensure that any optional parameters are not used (except for dlr-url to receive delivery acknowledgment on to



predefined URL). Following is the sample API call that will set the message priority for level 3 and Validity Period to 30 Minutes.

*Sample Format:*

```
http://api.instaalerts.zone/SendSMS/sendmsg.php?uname=XYZ&pass=ABC&send=mGage&dest=919980524000&msg=Testing&prty=3&vp=30
```

## 9 SENDING TEXT MESSAGE OF MORE THAN 160 CHARACTERS

The HTTP API support SAR (Segmentation and Reassembly - often called as concatenated SMS) of the message. The API will support maximum of 9 segmentations with max character limit of 1224. Should you need to send concatenated SMS, you need to set 'CONCAT' parameter value as 1. Following is sample API call:

*Sample Format:*

```
http://api.instaalerts.zone/SendSMS/sendmsg.php?uname=XXX&pass=XXX&send=XXXXXXX&dest=91XXXXXXXXXX&msg=(Message of more than 160 characters)&concat=1
```

## 10 SENDING UNICODE CHARACTER

Karix HTTP API has end to end support for Unicode characters. Should need to send a Unicode message, you need to set DCS value to 8 while the message text length should not exceed 70 characters. Following is sample call:

*Sample Format:*

```
http://api.instaalerts.zone/SendSMS/sendmsg.php?uname=XXX&pass=XXX&send=XXXXXXX&dest=91XXXXXXXXXX&msg=UnicodeString&udhi=1&dcs=8
```

When the UDHI value is set to 1 in API, it is required to tack the header before the message contents. If a single part Binary or Unicode message is sent, there is no need to set the udhi value and attach the header in the message. The API considers the default value of the udhi to zero.



***Sample Format for Single Part message:***

```
http://api.instaalerts.zone/SendSMS/sendmsg.php?uname=XXXXX&pass=XXXX&send=X
XXXXXXXXX&dest=91XXXXXXXXXX&msg=093909010917093E092E093E0020002009390
94B00200917092F093E&dc=8
```

In the above Sample message, it is a single part message and there is no need to tack header and set the udhi value.

The header includes the values of length of header in octet, Mobile source and destination ports and message parts information.

One way of sending concatenated SMS (CSMS) is to split the message into parts, and sending each part with a User Data Header (UDH) tacked onto the beginning. A UDH can be used for various purposes and its contents and size varies accordingly, but a UDH for concatenating SMSes look like this:

- Field 1 (1 octet): Length of User Data Header, in this case 05.
- Field 2 (1 octet): Information Element Identifier, equal to 00 (Concatenated short messages, 8-bit reference number)
- Field 3 (1 octet): Length of the header, excluding the first two fields; equal to 03
- Field 4 (1 octet): 00-FF, CSMS reference number, must be same for all the SMS parts in the CSMS
- Field 5 (1 octet): 00-FF, total number of parts. The value shall remain constant for every short message which makes up the concatenated short message. If the value is zero then the receiving entity shall ignore the whole information element
- Field 6 (1 octet): 00-FF, this part's number in the sequence. The value shall start at 1 and increment for every short message which makes up the concatenated short message. If the value is zero or greater than the value in Field 5, then the receiving entity shall ignore the whole information element.

***Sample Format for Multi-part message:***

***Part I:***

```
http://api.instaalerts.zone/SendSMS/sendmsg.php?uname=XXXXX&pass=XXXX&send=X
XXXXXXXXX&dest=91XXXXXXXXXX&msg=050003FA05010938094D0926094D092B094D
0938094D0926094D092B094D0938094D0926092B094D0938094D0926094D0917094D0
926094D092B094D0917094D0926094D092B094D0917094D0926094D092B094D09380
```

94D092B094D0938094D0926094D092B094D0938094D0926094D092B094D0938094D0926092B094D0938094D0926092B0938094D0926&udhi=1&dcs=8

**Part II:**

<http://api.instaalerts.zone/SendSMS/sendmsg.php?uname=XXXXX&pass=XXXX&send=XXXXXXXXX&dest=91XXXXXXXXXX&msg=050003FA0502094D092B094D0938094D0926094D092B094D0938094D09260940092B094D0939094D0938094D0926094D0915094D092B094D0939094D0915094D0938094D09260939094D092B094D0915094D0932094D0938094D0927094D092B094D0915094D0932094D0938094D0927094D092B094D0915094D0938094D0927094D092B094D0915094D0938094D0927094D092B094D0915094D0938&udhi=1&dcs=8>

**Part III:**

<http://api.instaalerts.zone/SendSMS/sendmsg.php?uname=XXXXX&pass=XXXX&send=XXXXXXXXX&dest=91XXXXXXXXXX&msg=050003FA0503094D0927094D092B094D0915094D0932094D0938094D0926094D091D094D092B094D0915094D0938094D0927094D092B094D0915094D0938094D0927094D092B094D0932094D0915094D0938094D0927094D092B094D0938094D0926094D0915094D0934094D092B094D0926094D0916094D092B094D0915094D0938094D0926092B094D0915&udhi=1&dcs=8>

**Part IV:**

<http://api.instaalerts.zone/SendSMS/sendmsg.php?uname=XXXXX&pass=XXXX&send=XXXXXXXXX&dest=91XXXXXXXXXX&msg=050003FA0505094D0938094D0926094D0915094D092B09410913092F094D0907092F094D0935093F091A094D0909092F094D0917094D092E0930094D092E094D0938094D0926003B0027091A094D092A094D&udhi=1&dcs=8>

**Part V:**

<http://api.instaalerts.zone/SendSMS/sendmsg.php?uname=XXXXX&pass=XXXX&send=XXXXXXXXX&dest=91XXXXXXXXXX&msg=050003FA0504094D091C094D0938094D0926094D092B094D0915094D0934094D092B094D0938094D0926094D0915094D0934094D092B094D0939094D0938094D0926094D092B094D0916094D0938094D09260932094D0915094D092B094D0939094D0938094D0926094D0915094D09320939094D092B094D0938094D0926094D0915094D09320939094D092B&udhi=1&dcs=8>

In above multi-part message, the udhi values should be set to 1 and the header should be tacked to every part message.

## 11 SENDING BINARY MESSAGE (160 CHARACTERS)

HTTP API supports for 8 bit binary message such as Ring tones, logos, Picture Messages. However, it needs to set certain parameter for the Binary message in HTTP API else the message would be treated as normal text message and may not receive as expected. The following parameters need to be set for sending binary messages:

- UDH should be set to 1
- DCS value should be 4 or 240 or 245 (based on the spec)
- Message should not exceed 280 characters

Following is the sample call:

*Sample Format:*

```
http://api.instaalerts.zone/SendSMS/sendmsg.php?uname=XXX&pass=AXX&send=XXXX
XXXX&dest=91XXXXXXXXXX&msg=BinaryMessageString&udhi=1&dcs=4
```

If the port number is defined in the header, the mobile device automatically identifies the message type whether it is a ringtone, picture, vCard, vCalendar etc.

There are few standard ports for mobile device that can be included in the messages to send specific message:

S.No	Type	Ports in Decimal
1	Ringtone	5505
2	CGI /CLI	5507
3	Picture Message	5514
4	Operator Logo	5506
5	VCard	9204
6	VCard Secure	9206
7	VCalendar	9205
8	VCalendar Secure	9207
9	E-Mail Notification	5512
10	DMCP (obsolete)	5508

## 12 SENDING BINARY MESSAGE OF MORE THAN 160 CHARACTERS

The Multi-part or concatenated SMS could also be referred to as a PDU Mode SMS. The number of parts that a multi-part or PDU mode SMS message may contain depends technically upon a header message but mostly upon the device sending or receiving the SMS and also upon the service provider.

In theory, the concatenated SMS may consist of up to 255 separate SMS messages that are concatenated in order to create a single long SMS message. Because of the nature of the SMS, the chance that these parts of the SMS message arrive in order is slim and therefore a strategy is implemented in order for the original long message to be reconstructed.

The Concat parameter can be set as 1 in the HTTP API to send more than 160 characters Binary message.

Following is the sample call:

### *Sample Format:*

```
http://api.instaalerts.zone/SendSMS/sendmsg.php?uname=XXX&pass=AXX  
&send=XXXXXXXX&dest=91XXXXXXXXXX&msg=BinaryMessageString&udhi=1&dc  
s=4&concat=1
```

## 13 SENDING WAP MESSAGES

HTTP API also supports the WAP (Wireless Application Protocol) messages where the following parameters need to be set:

### *Sample Format:*

```
http://api.instaalerts.zone/SendSMS/sendmsg.php?uname=XXX&pass=AXX  
&send=XXXXXXXX&dest=91XXXXXXXXXX&wapurl=http://mGagewap.com/mobile  
&waptxt=mGage
```

The API does not support the WAP concatenated SMS and can be sent max character limit of 160 characters.

## 14 SENDING INTERNATIONAL MESSAGES

HTTP API supports sending the International messages. To send the International messages through HTTP API, it is required to set the intl parameter as 1. The concatenated International message can also be sent through the API by adding the concat parameter.

There are some countries where the Sender ID is accepted only Numeric; hence, it needs to be taken care of while sending the International messages.

The following action would be taken when the Intl parameter is set as ON/OFF and the Destination Number is entered with different length:

S.No	API International Flag	Destination Number Length	Action
1	OFF/ON	Less than 9 digits or greater than 15 digits	Reject
2	OFF/ON	9 digits	Process to international
3	OFF	10 Digits	Process to Domestic and prefix 91
4	ON	10 Digits	Process to international
5	OFF/ON	11 digits starting with 0	Process to Domestic, remove 0 and prefix 91
6	ON/OFF	12 digits and starts with 91	Process to domestic
7	ON/OFF	All other cases for length between 12 to 15 digits only.	Process to international
8	Undefined	If mobile number is 12 digits starting with 91 send to domestic all other cases send to international	

Following is the sample call:

### **Sample Format:**

```
http://api.instaalerts.zone/SendSMS/sendmsg.php?uname=XXX&pass=AXX
&send=XXXXXXXXXX&dest=91XXXXXXXXXX&msg=MessageString&intl=1&concat=1
```

## 15 CALL BACK DELIVERY REPORT

Each time the SMS gateway attempts to deliver a SMS message to a mobile device, a Delivery Report (DLR) is generated against every MID (Message ID). This delivery report includes an indication if the message was successfully delivered and if not, the reason for failure.

The Delivery Report (DLR) for the messages sent through HTTP API can be downloaded from the Delivery Report frontend (<http://api.instaalerts.zone>) or using the DLR URL provided by Client.

Each time the SMS gateway attempts to deliver a SMS message to a mobile device, a Delivery Report (DLR) is generated against every MID (Message ID). This delivery report includes an indication if the message was successfully delivered and if not, the reason for failure.

The Delivery URL provided in this document is used to receive the Delivery Report against every MID generated on submission of message to Karix. This option is available for any customer who has opted to receive the Delivery Report on specifically on HTTP.

For this purpose an API needs to be provided to Karix to receive the API calls on delivery reports. Against this API call Karix can pass parameters as requested

Parameters supported by the API call include

S.No	Parameters	Type	Description
1	SID	bigint (25)	This Unique Message ID that is provided by Karix for tracking purpose against every submission of SMS for delivery done to Karix.
2	DEST	bigint (15)	The number to which the SMS was intended to be terminated.
3	STIME	timestamp	This is a time stamp from Karix Servers against the submission of message for the indicated MID.(YYYY-MM-YY HH:MM:SS)
4	DTIME	timestamp	This is a time stamp received by Karix from the subscribers terminating operator against the submission of message for the indicated MID. (YYYY-MM-YY HH:MM:SS)
5	STATUS	Varchar (30)	This indicates the final status of the message, the description of the status is also provided below in this document.
6	REASON	tinyint (4)	This is the reason for the final status received from the terminating operator on the delivery status.

When a Message is sent by the Client through HTTP API or XML API, a MID is generated and sent to the Client against the message submission. The Application at Client's end needs to capture the MID after sending the message and then should keep listening to the DLR API to capture the DLR against the MID sent.

## 16 SAMPLE URL FORMAT

The DLR URL needs to be provided by Client so that the same can be configured in the Application at Karix's Server to send the Delivery Report. A sample of the URL could be in the below format

### *Sample Format:*

```
CLIENTSERVERIP/PATH?sid=XX&dest=XX&stime=YYYY-MM-YY
HH:MM:SS&dtime=YYYY-MM-YY HH:MM:SS&status=XXX&reason=XXXX
```

### *Sample API Call made by Karix:*

```
CLIENTSERVERIP/PATH?sid=21232324243432424234&dest=919886430811&stime=2011-04-21 12:32:04&dtime=2011-04-21
12:32:16&status=001&reason=DELIVRD
```

## 17 POLLING DELIVERY REPORTS:

When a Message is sent by the Client through HTTP API or XML API, a uniqueID is generated viz MID and sent to the Client on submission. Karix will make delivery attempt to deliver the SMS, on completion the SMS gateway will receive a Delivery Report (DLR) against every MID (Message ID). This delivery report includes an indication if the message was successfully delivered and if not, the reason for failure.

The Delivery URL provided in this document is used to poll Delivery Report if any on scheduled intervals. This option is available for any customer who has opted to poll Delivery Report specifically on HTTP/HTTPS. For this purpose an API call needs to be submitted to Karix to poll for any available delivery reports which would include the below parameters:

S.No	Parameter	Type	Description
1	uname*	string	Username Assigned to the account
2	pass*	string	Password Assigned to the account
3	type*	string	Preferable format in which response is expected. Plain for HTTP and XML for XML



#### Sample Format for HTTP:

```
http://api.instaalerts.zone/SendSMS/dlr_status.php?uname=XXX&pass=XXX&type=plain
```

#### Sample Format for XML:

```
http://api.instaalerts.zone/SendSMS/dlr_status.php?uname=XXX&pass=XXX&type=xml
```

On submitting the above request Karix will check if for any available delivery reports that could be provided and submit the same as an echo response with the below parameters

S.No	Parameters	Type	Description
1	MID	bigint (25)	This Unique Message ID that is provided by Karix for tracking purpose against every submission of SMS for delivery done to Karix.
2	STIME	timestamp	This is a time stamp from Karix Servers against the submission of message for the indicated MID.
3	DTIME	timestamp	This is a time stamp received by Karix from the subscribers terminating operator against the submission of message for the indicated MID.
4	STATUS	Varchar (30)	This indicates the final status of the message, the description of the status is also provided below in this document.
5	REASON	Varchar (30)	This is the reason for the final status received from the terminating operator on the delivery status.
6	SENDERID	Varchar(15)	Source address for the message that was received by the MS.
7	DEST	bigint (15)	The number to which the SMS was intended to be terminated.

## 17.1 Key Notes for polling DLR

- Delivery reports are available for up to 3 hours from receipt.
- Each API call will have upto 100 delivery reports
- It is recommended that in case there are no delivery reports please restrict API calls to one every 5 minutes.

## 17.2 Sample response on HTTP for tilde separated:

In a HTTP API request each of the parameters is separated by “~” (tild). A sample of the same is provided below

```
mid~stime~dtime~status~reason~senderid~dest
2111071507000174250~2011-07-15 07:00:00~2011-07-15
07:00:17~DELIVRD~NULL~LM-mGage~919243201476
2111071507000574530~2011-07-15 07:00:02~2011-07-15
07:00:19~DELIVRD~NULL~919243201476~919686555555
2011071507010111690~2011-07-15 07:00:58~2011-07-15
07:01:18~DELIVRD~NULL~LM-MGAGE~919880015015
2011071507010111700~2011-07-15 07:00:57~2011-07-15
07:01:23~DELIVRD~NULL~LM-MGAGE~919591996400
2011071507060112020~2011-07-15 07:06:00~2011-07-15
07:06:20~DELIVRD~NULL~LM-MGAGE~919591996400
2011071507060112010~2011-07-15 07:06:00~2011-07-15
07:06:20~DELIVRD~NULL~LM-MGAGE~919880015015
2111071507100207100~2011-07-15 07:09:59~2011-07-15
07:10:16~DELIVRD~NULL~LM-mGage~919243201476
2111071507100407560~2011-07-15 07:10:01~2011-07-15
07:10:18~DELIVRD~NULL~919243201476~919686555555
2011071507110112370~2011-07-15 07:10:58~2011-07-15
07:11:17~DELIVRD~NULL~LM-MGAGE~919880015015
2011071507110112380~2011-07-15 07:11:00~2011-07-15
07:11:27~DELIVRD~NULL~LM-MGAGE~919591996400
2011071507160112710~2011-07-15 07:15:59~2011-07-15
07:16:19~DELIVRD~NULL~LM-MGAGE~919880015015
```

## 17.3 Sample response on HTTP XML response:

In a XML API request each of the parameters is tagged, a sample of the same is provided below

```
<dlr><report mid="2111071508050395580" stime="2011-07-15 08:04:59"
dtime="2011-07-15 08:05:16" status="DELIVRD" reason="NULL"
senderid="919212355638" dest="919972648090"/><report
mid="2111071508050395650" stime="2011-07-15 08:04:59" dtime="2011-
07-15 08:05:17" status="DELIVRD" reason="NULL"
senderid="919212355638" dest="919704000100"/><report
mid="2111071508050396140" stime="2011-07-15 08:04:59" dtime="2011-
07-15 08:05:17" status="DELIVRD" reason="NULL"
senderid="919212355638" dest="919900093039"/><report
mid="2111071508050295160" stime="2011-07-15 08:04:59" dtime="2011-
07-15 08:05:16" status="DELIVRD" reason="NULL"
senderid="919212355638" dest="919845656000"/><report
mid="2111071508050295340" stime="2011-07-15 08:04:59" dtime="2011-
07-15 08:05:16" status="DELIVRD" reason="NULL"
senderid="919212355638" dest="919704100100"/><report
mid="2111071508050395510" stime="2011-07-15 08:04:59" dtime="2011-
07-15 08:05:17" status="DELIVRD" reason="NULL"
senderid="919212355638" dest="919740034444"/><report
mid="2111071508050395640" stime="2011-07-15 08:04:59" dtime="2011-
```

```
07-15 08:05:16" status="DELIVRD" reason="NULL"
senderid="919212355638" dest="919650656262"/><report
mid="2111071508050395610" stime="2011-07-15 08:04:59" dtime="2011-
07-15 08:05:16" status="DELIVRD" reason="NULL"
senderid="919212355638" dest="919771416994"/><dlr>
```

## 18 STATUS CODES AND REASON RECEIVED ON THE URL

CODE	REASON	DESCRIPTION
<b>I) FINAL STATUS</b>		
<b>-01</b>	New Error Code	This status displays when the error code provided by the subscribers terminating operator is not mapped with the existing status.
<b>000</b>	Sub-SMSC	The message is on SMSC queue i.e. the message has been inserted into the SMSC database but the status of the message is yet to be received.
<b>001</b>	DELIVRD	Successfully delivered.
<b>002</b>	FAILED	The message is permanently failed due to CallBarred, Error in Destination Number, Error in TeleService Provider etc.
<b>004</b>	NDNC_Failed	Failed Due to DND Registration
<b>044</b>	Promo_Blocked	This error is displayed when IUC charges are not active for your account.
<b>005</b>	Blacklist	Black-listed number. This list of numbers is provided by the customer and includes numbers of CEO etc. A message will never go to a number in the black-list.
<b>006</b>	Whitelist	This error is received when a Opt-In account sends messages to a non white listed number
<b>007</b>	Invalid Series	This error is shown when a number series in the correct number format is invalid.
<b>008</b>	Prepaid Reject	This error is shown when the messages are rejected due to insufficient credits.
<b>009</b>	Night_Expiry	These messages have not been processed because of legal restrictions of sending messages late hours.
<b>099</b>	Night_Purge	These are promotional messages submitted between 9PM and 12AM which are not processed on request.
<b>031</b>	EXP-AbsSubs	The message is rejected because there was no paging response, the IMSI record is marked detached, or the MS is subject to roaming restrictions.
<b>032</b>	EXP-MEM-EXCD	Message rejected because the MS doesn't have enough memory.

<b>033</b>	EXP-NW-FAIL	Message rejected due to network failure.
<b>034</b>	EXP-NW-TMOUT	Message rejected due to network or protocol failure.
<b>035</b>	EXP-SMS-TMOUT	Message rejected due to network or protocol failure.
<b>036</b>	EXP-HDST-BUSY	The message is rejected because of congestion encountered at the visited MSC.
<b>037</b>	EXP-MSG-Q-EXD	Message queue exceeded when there are too many messages for one number. SMSC can deliver only a particular number of messages to a mobile number. If there are more messages, they get this error code till the queue clears.