



HTTP API (Application Programming Interface)

The Business Centre 8, Gemsbok Lane Bryanston, 2125, RSA.

P.O.BOX 2187 Ikeja Lagos Nigeria.

Web: <http://phanerus.com>
Email: support@phanerus.com
Help desk: (27) 0116123138

Updates and Changes	
Last updated	30 th January 2011
Updated by	Chris Sardius

In obtaining this document you have taken the first step towards enabling your business with our premiere mobile billing processor. The aim of this document is to show you how to integrate **phanerus shortcode or longcode services** within your own systems. So please follow this guide as closely as possible and direct any questions you may have to our Support and Development service available via: support@phanerus.com

There are three simple steps to a world of possibilities:

- 1: Fill out your variable information within the online API.
- 2: Create your server-side scripts, and integrate them.
- 3: Advertise your service and view your incoming messages and/or Account Balance in your phanerus user area <http://sms.phanerus.com/user/login>, under **shortcode or longcode services**.

API FUNCTIONALITY

Short code: A short telephone number / code, usually 4 to 8 digits long, that a user sends a text message to in order to communicate with your service. Please contact us or refer to short code services in phanerus user area to check which code is in use.

Username: The unique username that identifies your phanerus account which is the same as your email address . E.g. 'user@phanerus.com'.

Message: The text body of an SMS message.

Trigger Code: The first keyword in the SMS body which tells our system how to process the message. Can be: ask, chat, interact, talk, or your own dedicated keyword (log into phanerus under shortcode or longcode services to set one up).

eKey: The unique encrypted identifier key used to ensure the POST originated from the certified source. This is assigned to the client when they have completed payment and the registration process. You can see this in your phanerus user area. Click start short code campaign.

The Shortcode or longcode service on phanerus can be used for delivery of your services (e.g. Alerts & subscriptions), and campaigns (e.g. communication with your existing database).

You will need:

1. Good programming knowledge to create a script that can respond to POST requests.
2. A Short code campaign enabled on your account.
3. To restrict access to the script from only the mBILL server IP.
4. To set up your phanerus account to POST to your script URL.
5. To have the Short code service activated on your account.

Max. Length of SMS Header: 11 Characters.

Common Error Codes:

101: Duplicate post back for message ID.

102: Binary transaction requested, but UDH not specified.

103: An invalid eKey has been set and posted to us.

104: Invalid details. Please check and try again.

415: Invalid username set, please check the "cc" variable.

401: Invalid or unrecognized action variable set.

Process of a Transaction:

SDR works predominantly through the use of HTTP POST's. Many crucial (variable) components of the POST are detailed later in this document.

However this page focuses on the transaction process, as well as the different "action" variables. Within the HTTP POST, you'll choose your action value to be one of the below.

mpush_sdr_message and sdr_response actions are used by our server (in the case of mpush_sdr_message) and sdr_response from your server when responding to an mpush_sdr_message. The sdr_gateway action is used to initiate a transaction.

mpush_sdr_message & sdr_response

1. User sends a SMS message to our short-code in the format:

'[TriggerCode] [username] [Message]' Eg. 'ask user@phanerus.com this is a test'

2. Our server receives the message and sends the Message details to your requested URL to process as required. (mpush_sdr_message)

3. You receive and store the details for reference - and for future message processing.

4. You build the response message and send us the required information in a POST back to our server. (sdr_response).

5. Message is returned to the end-user's handset.

Step 1 :: Your reception

This step details all of the information you need regarding variables pushed to and from our servers. We send you five variables: ACTION, ID, NUMBER, NETWORK and MESSAGE. Only allow access to your script from our server IP's as listed here to avoid malicious attempts to post to your server.

IP Block: 64.202.107.96, 64.202.107.97, 64.202.107.98, 64.202.102.182, 64.202.102.183, 64.202.102.184, 64.202.102.185, 64.202.102.186, 64.202.102.187, 64.202.102.188, 64.202.102.189, 64.202.102.190, 64.202.102.191

Step 2 :: Post-back

Step two is the process where you POST back to us. All posts to us (both Gateway and Response) must be sent to our POST URL . You must include the following variables with values: ACTION, ID, NUMBER, NETWORK, MESSAGE, RESPONSE, VALUE and your EKEY. For your convenience, these variables are explained in more detail later on in this document.

Step 3 :: Security

Step three is simply details of your eKey (encrypted key). This is an essential component of any successful POST from you to our system. Without this your message will be rejected, so please take care to ensure you have the right eKey within your POST.

The POST Messages

To configure your HTTP responder(s) you will need to process the following post variables. The values sent/expected within these variables are given below.

mpush_sdr_message (from phanerus to you)

This POST is sent from our server to yours, and will contain details of an (the initial) incoming SMS message.

Variable	Content
Action	mpush_sdr_message
ID	The unique message number
Number	The originating telephone number
Network	The originating network.
Message	The contents of the message
Shortcode or longcode	The incoming shortcode or longcode of the message.
Country	The 2 digit ISO country code (e.g. UK, US, ES).

sdr_response (your response to our mpush_sdr_message) This POST is sent from your server to ours, and contains details of the outgoing message to be sent to the handset user.

Variable	Content
Action	sdr_response
ID	The unique message number.
Number	The originating telephone number.
Network	The originating network.
Message	The contents of the message Note this is always in the form message=ask+{username}+message
response	Your message response (maximum of 160 characters).
ekey	Your encrypted key.

Type	URL to post message to
HTTP	http://onecast.smsdam.com/?mob=sdr_response

Example Source Code (PHP Responder to posts from your server)
For your convenience, please see below for an example Responder Script:

```
<?php
/** @brief eKey found within user area */
$strEkey = 'cdbaea40d2bbc62a56d4aec3c9f8';

/** @brief Array containing valid server IP addresses. */
$arrAllowedServers = array('64.202.107.96', '64.202.107.97', '64.202.107.98', '64.202.102.182', '
64.202.102.183', '64.202.102.184', '64.202.102.185', '64.202.102.186', '64.202.102.187', '
64.202.102.188', '64.202.102.189', '64.202.102.190', '64.202.102.191');

/** @brief Output variable used to store the output of the script. */
$strOut = null;
// Process the incoming post request.
// Check that the transaction originated from a phanerus server.

if(in_array($_SERVER['REMOTE_ADDR'],$arrAllowedServers)){
switch($_POST['action']) {
case 'mpush_sdr_message': {
/** @brief phanerus server URL for the return post. */
$strPostUrl = 'http://onecast.smsdam.com/?mob=sdr_response;
/* Setup return post variables ready for the post string to be constructed.*/

$strPostReq = 'action=sdr_response&id=%d&number=%s&network=%s&message=%s'.
'&response=%s&ekey=%s';

$intId = (int) $_POST['id'];
$strNumber = $_POST['number'];
$strNetwork = $_POST['network'];
$strMessage = $_POST['message'];
/** @brief Generate the return post string. */
```

```

$strPostReq = sprintf($strPostReq,
    $intId,
    $strNumber,
    $strNetwork,
    $strMessage,
    $strResponse,
    $strEkey
);

// Setup curl to initiate post back to phanerus.
$resCurl = curl_init();
curl_setopt($resCurl,CURLOPT_URL,$strPostUrl);
curl_setopt($resCurl,CURLOPT_POST,1);
curl_setopt($resCurl,CURLOPT_POSTFIELDS,$strPostReq);
curl_setopt($resCurl,CURLOPT_RETURNTRANSFER,1);

$strBuffer = curl_exec($resCurl);
curl_close($resCurl);
// Check to see if return post is a success.
if(0==(strpos($strBuffer,'SUCCESS'))){
    $strOut = 'OK';
    // Process is now complete.
} else {
    $strOut = "Error: $strBuffer";
    /* If required, ensure that your script logs the error for future reference. */
}
break;
}
default: {
    /** @note This error is generated if the SDR action variable is incorrect. */
    $strOut = 'Error: Unrecognised SDR Action!';
    break;
}
} else {
    $strOut = "$_SERVER[REMOTE_ADDR] : Access Denied! Unauthorised server IP.";
}
/** @brief Once the script has been run output the result. */
echo $strOut;
?>

```