

<http://web-expert.gr>

# SMS NOTIFY 3.x SERIES DEVELOPER'S GUIDE

**How to integrate your gateway easily!**

**H**ello, first of all I would like to thank you for your participation to my project. If you want, you can send your gateway files to [info@web-expert.gr](mailto:info@web-expert.gr) to include it to main release package. For any question don't hesitate to contact me.

Let's Start!

Firstly you need the PHP basic and the notepad++ (google it)

## Display the errors: Blank page

Go to whmcs root at configuration.php and add

```
$display_errors = true;
```

Or

WHMCS>Setup>General Settings>(Tab) Other> Tick Display Errors

## Create a PHP File

All gateway addons files must be located to

*/modules/addons/sms\_notify/smsproviders (whmcs v5.x)*

Create a php file and give a unique name without special chars or spaces

If you need to have more than one file create a folder inside the 'smsproviders'

Look the example **mygateway.php** and place it at "smsproviders" folder. We give all sms gateways integrations unlocked, so you have the change to look the code.

Now open *mygateway.php* to see the template code (vars, functions and comments)

## What you can do with the SMS Notify Integration

- You can set custom params & provide info about gateway
- You can sendSMS (to one or many phones & scheduled)
- You can get the SMSstatus of sent SMS (callback updated supported)
- You can get the Balance of account

## Available Functions

- **\_gatewaydetails()** returns array with info and custom fields
- **\_sendsms(\$params)** returns array smsid / error
- **\_getSmsBalance(\$params)** returns total credits / error
- **\_getsmsstatus(\$params)** returns sms status code or error

**If a feature is not supported** example the `_getsmsstatus()` delete the function completely from your file. Required functions are only the `_gatewaydetails()` & `_sendsms()`

## Some Defined Functions (API) you can use:

**getRemoteData(\$url,\$fields=null,\$headers=array(),\$noCURL=false)**

*\$url* field is required. If you pass only the URL it will use GET action.

*\$fields* should contain the fields of POST action. Can be string or array

*\$headers* you can push senders in array format. `array('Content-Type: application/json')`

Response: `@array=response,error`

**unicode(\$str,\$mode='') returns Unicode\_string**

*\$mode* values 'UTF-16', 'UCS-2BE', 'UCS-4BE' (or empty)(default)

**smscut(\$str,length)**

*it uses mb\_substr if exists. Returns a smaller string*

`logger($action,$requeststring,$responsedata,$processeddata,$vars=array())`

*params are same with logModuleCall but without the first parameter (\$module)*

[http://docs.whmcs.com/Provisioning\\_Module\\_Developer\\_Docs#Module\\_Logging](http://docs.whmcs.com/Provisioning_Module_Developer_Docs#Module_Logging)

## Debugging Tip

You can send as text an array using the

```
$debug_text=print_r($params,true);  
  
mail("my@mail.com",$debug_text,"Debug sendsms params");
```

## Create status and error handler at `_getsmsstatus()`

### 1) SMS Statuses Numbers

Our plan is to have many gateways to our project and we tried to have a specific format to statuses because every gateway has different way to represent the SMS status.

So these are the values that represent 2 permanent and 2 not permanent statuses. Total 4. The not permanent statuses when the admin/user visits the "History Log" page then the script tries to get the status again and If is changed it will update the log sql table.

### Allowed Statuses

- **0** Not Delivered/Failed/ Error (Permanent)
- **1** Delivered Successfully (Permanent)
- **2** Sent Successfully but Pending (Not Permanent)
- **3** Unknown (Not Permanent) \*

Most gateways have many other statuses, so you need to group it to above 4 categories (0 to 3), so let's say that the `$sms_status_text` has the text sms status of a sms message that we are extracted after the http call

```

switch($sms_status_text){
    case 'Failed':
    case 'Not Delivered':
        $sms_status_num=0;
        break;
    case 'Delivered':
        $sms_status_num=1;
        break;
    case 'Sent':
    case 'Pending':
    case 'Buffered':
        $sms_status_num=2;
        break;
    case 'Unknown':
        $sms_status_num=3;
        break;
    default:
        $sms_status_num=3; #or =0;
}

```

## 2) Errors

The same think above you can do with errors numbers. You can create a php switch and give a text description to pass the error to returning value <<\$values["error"]>>. Don't forget to pass the errors to value it will help you to find the bugs of your add-on, also it will let know the user what happened and sms is not processed

## Callback

If your gateway supports callback sms status updates then it's easy to use this feature. Just look the callback/mygateway.php example. If not supports this feature delete the file

In most gateways the CallbackURL is defined inside the gateways control panel (like Clickatell)

## Extract the information you need from response var (\$data["response"])

Example:

You must first know what response text is returned after http call. For example the yuboto.com (Bulk-SMS), if success, returns the following format

Sent:xxxx-xxx-xxxx-xxx-xxxx (after : is the SMS\_ID, that we want!)

There are 2 simple ways!

### 1)

```
$response= explode(":",($data["response"]),2);
```

```
$response[0] #Sent
```

```
$response[1] #SMID (done!)
```

### Add a check first to know if is sent

```
If($response[0]=='Sent') {  
    $values["smsid"]= $response[1];  
}else{  
    $values["error"]= $data["response"];  
}
```

### 2)

```
if (substr($data["response"], 0, 4) == "Sent"){  
    $values["smsid"]=str_replace('sent:',",$data["response"]);  
}else{  
    $values["error"]= $data["response"];  
}
```

### or

```
if (substr($data["response"], 0, 4) == "Sent"){  
    $values["smsid"]= substr($data["response"],5);  
}
```

```
}else{  
    $values["error"]= $data["response"];  
}
```

The same way you can handle the error messages that are returned

### **For Beginners =>PHP Resources**

<http://php.net/manual/en/function.substr.php> (sub /cut string based on length)

<http://php.net/manual/en/function.explode.php> (string to array)

<http://php.net/manual/en/function.str-replace.php> (find & replace)

<http://php.net/manual/en/function.strlen.php> (string length)