



SMStrend

PHP SDK

INTRODUCCIÓN

Los servidores SMStrend ponen a disposición de los desarrolladores un sistema de comunicación HTTP para enviar y recibir mensajes SMS a través del gateway SMStrend y permiten recuperar todos los datos sobre el estado de los mensajes y el histórico SMS.

CONFIGURACIÓN DE LA LIBRERÍA

Modificar login (nombre usuario) y password (contraseña) en el fichero config.php:

```
<?php

define('SMSTREND_HOSTNAME','www.smstrend.net');
define('SMSTREND_USERNAME','your_SMStrend_login');
define('SMSTREND_PASSWORD','your_SMStrend_password');
define('SMSTREND_DEFAULT_PORT',80);

?>
```

ENVÍO DE MENSAJES SMS

La clase `SMStrend_SMS` (file `sendsms.php`) contiene todos los datos necesarios para enviar un mensaje SMS a través del gateway `SMStrend`.

Una vez creado un objeto de tipo `SMS`, es necesario establecer:

- tipo de mensaje: `SMSTYPE_GOLD` o `SMSTYPE_GOLD_PLUS` (atributo `sms_type`);
- destinatarios: número de teléfono de los destinatarios en formato internacional (ex: +34610123456); para añadir un número a la lista de los destinatarios, utilizar el método `add_recipient($string)`
- mensaje: atributo `message`, establece el texto del mensaje que se quiere enviar, máximo 160 caracteres por un síngulo SMS, máximo 1000 caracteres por un SMS largo (concatenado). Si el texto del SMS es superior a 160 caracteres, el sistema va a enviar un SMS concatenado compuesto por "n" SMS de 153 caracteres cada uno. Atención: los siguientes caracteres se cuentan como caracteres dobles: `^ { } \ [~] | € .`; es posible utilizar el método `length()` para conocer el número total de caracteres del SMS, y el método `count_smss()` para conocer el número de SMS necesarios para efectuar el envío;
- remitente: en caso de SMS con remitente personalizado (`GOLD PLUS`). El remitente personalizado puede ser un texto de máximo 16 caracteres numéricos para un número internacional o de 11 caracteres alfanuméricos para un texto (ex: "+123456789012345" o "EstaEmpresa"), atributo `sender`.

Opcionalmente, es posible introducir:

- fecha y hora del envío SMS programado (método `set_scheduled_delivery($unix_timestamp)`);
- una secuencia alfanumérica asociada al envío, indispensable para obtener todas las informaciones que se refieren al estado de los mensajes (atributo `order_id`). La longitud máxima es de 32 caracteres, todos los caracteres excedentes no serán considerados.

Ejemplo:

```
require('../sendsms.php');

$sms = new SMStrend_SMS();
$sms->sms_type = SMSTYPE_GOLD;
$sms->add_recipient('+34610123456');
$sms->add_recipient('+34610123456');
$sms->message = 'hello world!';
$sms->sender = 'test';
$sms->set_immediate(); // or sms->set_scheduled_delivery($unix_timestamp)
$sms->order_id = '999FFF111';
echo 'About to send a message ' . $sms->count_smss() . ' SMSs long ' ;
echo 'to ' . $sms->count_recipients() . ' recipients </br>';
if ($sms->validate()) {
    $res = $sms->send();
    if ($res['ok']) {
        echo $res['sentsmss'] . ' SMS sent, order id is ' . $res['order_id'] . ' </br>';
    }
}
```

```
} else {  
    echo 'Error sending SMS: '.$sms->problem().' </br>';  
}  
} else {  
    echo 'invalid SMS: '.$sms->problem().' </br>';  
}
```

ESTADO DE LOS MENSAJES

Después del envío de los mensajes es posible pedir el estado de cada SMS, utilizando la función `smstrend_get_message_status($order_id)`. Especificando el id del SMS, se pide al servidor el resultado del envío y el servidor devuelve una lista de objetos `SMStrend_SMS_STATUS`; cadauno de los objetos contiene:

- el número de teléfono del destinatario (atributo `recipient`);
- el estado del envío (atributo `status`);
- fecha y hora de efectiva recepción del mensaje, si el SMS ha sido correctamente recibido por el destinatario. El atributo `dt_received` contiene la fecha/hora en formato secuencia, el metodo `get_sms_received_timestamp()`, en cambio, devuelve la fecha/hora como unix timestamp; es también posible utilizar la función `strtotime()` para obtener una asociación como en <http://it.php.net/manual/en/function.strptime.php>, la secuencia de formato utilizada es `SMSTREND_DATE_TIME_FORMAT` (véase ejemplo).

El número de objetos `SMStrend_SMS_STATUS` devueltos está contenido en la asociación misma y tiene la secuencia `count` como asociación (véase ejemplo abajo).

El atributo `status` puede tener los siguientes valores:

<code>SCHEDULED</code>	// retrasado, todavía no enviado
<code>SENT</code>	// enviado, no espera delivery
<code>DLVRD</code>	// el SMS ha sido correctamente recibido
<code>ERROR</code>	// error en la entrega del SMS (ex: número no existente)
<code>TIMEOUT</code>	// el operador non ha proporcionado informaciones sobre el SMS dentro de 48h
<code>TOOM4NUM</code>	// demasiados SMS por el mismo destinatario en las últimas 24h
<code>TOOM4USER</code>	// demasiados SMS enviados por el usuario en las últimas 24h
<code>UNKNPFX</code>	// prefijo SMS inválido o desconocido
<code>UNKNRCPT</code>	// número de teléfono del destinatario inválido o desconocido
<code>WAIT4DLVR</code>	// mensaje enviado, en espera de delivery
<code>WAITING</code>	// en espera, todavía no enviado
<code>UNKNOWN</code>	// estado desconocido

Si no se quiere implementar una gestión particular de los eventuales errores de envío, el método `is_error()` devuelve `true` si el mensaje está en error, `false` de lo contrario.

El método `str_status()` proporciona una descripción textual del estado.

Ejemplo:

```
require('../sms_status.php');

$status = smstrend_get_message_status('1F11DC4A77A8F');
if ($status['ok']) {
    for ($i=0;$i<$status['count'];$i++) {
        if ($status[$i]->is_error()) {
            echo 'Error sending sms to '.$status[$i]->recipient.': ';
            echo $status[$i]->str_status();
        } else {
            echo 'Message to number '.$status[$i]->recipient.': ';
            if ($status[$i]->status == 'DLVRD') {
                echo 'correctly delivered on ';
                echo date('l ', $status[$i]->get_sms_received_timestamp());
                echo ' at ';
                echo date('H:i', $status[$i]->get_sms_received_timestamp());
            } else {
                echo $status[$i]->str_status();
            }
        }
        echo '<br/>';
    }
} else {
    echo 'Request failed: '.$status['errcode'].' - '.$status['errmsg'];
}
```

HISTÓRICO DE LOS MENSAJES

La función `smstrend_get_sms_history($from,$to)` pide al servidor SMStrend la lista de los SMS enviados en un intervalo de tiempo específico; el servidor devuelve una secuencia de objetos `SMStrend_SENT_SMS`. Los parámetros `$from` y `$to` son unix timestamp.

Ejemplo:

```
require('../sms_history.php');

$history = smstrend_get_sms_history(time()-(7*24*60*60),time());
if ($history['ok']) {
    for ($i=0;$i<$history['count'];$i++) {
        echo 'SMS id '.$history[$i]->order_id.' sent from '.$history[$i]->sender;
        echo ' to '.$history[$i]->recipients_count.' recipients<br/>';
    }
} else {
    echo 'Request failed: '.$history['errcode'].' - '.$history['errmsg'];
}
```

CRÉDITO SMS DISPONIBLE

La función `smstrend_get_credits()` pide al servidor SMStrend la disponibilidad de crédito SMS; el servidor devuelve una secuencia de objetos `SMStrend_CREDIT`. Cada objeto `SMStrend_CREDIT` contiene: el tipo, la nación (null si crédito internacional) y la cantidad de crédito disponible.

Ejemplo:

```
require('../credits.php');

$credits = smstrend_get_credits();
if ($credits['ok']) {
    for ($i=0;$i<$credits['count'];$i++) {
        if (!$credits[$i]->is_international()) {
            echo 'You can send '.$credits[$i]->availability.' smss';
            echo ' of type '.$credits[$i]->credit_type;
            echo ' in '.$credits[$i]->nation.' <br>';
        } else {
            if ($credits[$i]->credit_type == 'EE') {
                echo 'You can send '.$credits[$i]->availability;
                echo ' smss in foreign countries <br>';
            }
            if ($credits[$i]->credit_type == 'NL') {
                echo 'You can perform '.$credits[$i]->availability;
                echo ' lookup requests <br>';
            }
        }
    }
} else {
    echo 'Request failed: '.$status['errcode'].' - '.$status['errmsg'];
}
```

VERIFICACIÓN DE LA VALIDEZ DE LOS NÚMEROS DE TELÉFONO (NUMBER LOOKUP)

La verificación de la validez es una operación que permite conocer el estado de existencia de un particular número móvil y, en el caso de que sea válido, la nación y la compañía telefónica.

La operación se realiza a través de la función

`smstrend_lookup_number($recipient)`, el objeto `SMStrend_LOOKUP_RESULT` así obtenido contendrá todos los detalles relativos al número solicitado.

Ejemplo:

```
require('../number_lookup.php');

$res = smstrend_lookup_number('+34610123456');
if ($res['ok']) {
    if ($res->valid) {
        echo 'Phone number ' . $res['lookup']->phone_num;
        echo ' is a valid ' . $res['lookup']->operator . ' mobile number';
        echo ' hosted in ' . $res['lookup']->nationName;
    } else {
        echo 'Invalid phone number';
    }
} else {
    echo 'Request failed: ' . $status['errcode'] . ' - ' . $status['errmsg'];
}
```