



**SmsBr – Soluções em torpedos para celulares**  
**HTTP Integração**

Itens do documento

1	Introdução
2	Enviando Mensagens (SMS)
3	Retorno status (HTTP)
4	Consulta de Saldo
5	Retorno automático de STATUS

Documento revisão 2.2

## 1. Introdução

Plataforma de envios SmsBr oferece vários métodos para enviar e receber mensagens SMS. Este documento contém as especificações para os seguintes métodos:

- Enviar mensagens usando HTTP POST
- Receber mensagens usando HTTP POST
- Receber relatório de entrega HTTP POST

## 2. Enviando Mensagens (SMS)

### *HTTP POST – Envio único*

A URL de destino para a instrução dos dados:

<https://smsbr.com.br/enviosms.php>

Independente da linguagem de programação o uso desta url é válido, basta enviar as seguinte variáveis como um POST:

### **Parâmetros Especificações**

<b>AUTENTICAÇÃO</b>	chaveAPI	Cliente chave de acesso localizado no topo da página do cliente.
	usuarioNome	Cliente username, login utilizado para o sistema SmsBr
<b>MENSAGEM</b>	numeroTel	Numero do celular de destino, formato internacional (558499009900)
	mensTexto	Mensagem de texto máximo 160 caracteres.
	id_propio	Identificador da mensagem (Opcional)
	flash	Tipo de mensagem aceita valores SIM/NAO

Durante este envio nossos servidores lhe retorna o APIMsgID, que pode ser armazenado para posterior uso, exemplo completo de envio usando PHP:

<?php

```
function sendSMSBR($chaveAPI,$usuarioNome,$numeroTel,$mensTexto,$id_propio){
    #-----[Envio dos dados]-----
    $sessao_curl = curl_init();
    curl_setopt($sessao_curl, CURLOPT_URL, "https://smsbr.com.br/enviosms.php");
    curl_setopt($sessao_curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($sessao_curl, CURLOPT_POST, 1);
    curl_setopt($sessao_curl,CURLOPT_POSTFIELDS,
        "chaveAPI=".$chaveAPI."&usuarioNome=".$usuarioNome."&numeroTel=".$numeroTel."&mensText
to=".$mensTexto."&id_propio=".$id_propio."&flash=NAO");
    $resultado = curl_exec($sessao_curl);
    curl_close($sessao_curl);
    $aux_id = explode(":",$resultado);

        if ($aux_id[0]=="sucesso"){
            return $aux_id[1];
        }else{
            return $resultado;
        }
    }

#-----[Recebendo dados via $_POST]-----
$chaveAPI = "482414404"; #Chave API do Usuario
$usuarioNome = "adelmo"; #Nome de usuario Login
$numeroTel = "5584987056005"; #Celular de destino Ex. (558499998888)
$mensTexto = "Mensagem de Teste SMSBR"; #Mensagem

$envio = sendSMSBR($chaveAPI,$usuarioNome,$numeroTel,$mensTexto,$id_propio);

echo $envio;

?>
```

Os exemplos acima fazem o envio das mensagens no formato alpha numéricos, o formato do numero do destinatário deve ser formato internacional (558499009900), 55-Pais, 84-DDD, 99009900-Celular de destino.

## Flash SMS.

Para o envio do Flash SMS nossa API tem um parâmetro opcional para indicar que a mensagem deve ser enviada como um Flash SMS. Este parâmetro chama-se “**flash**” e aceita os valores SIM (como Flash SMS) ou NAO (como SMS normal). Para efeitos de compatibilidade, caso o parâmetro não seja informado na requisição a mensagem será considerada um SMS normal.

### ***INTEGRAÇÃO VIA RUBY ON RAIL – Envio único***

Para enviar mensagens via Ruby on Rails, deve-se utilizar dos seguintes trechos de código em seu script, geralmente em alguma ação de seu controlador:

```
#requires necessarios, geralmente adicionado no inicio do arquivo require
"uri"
require "net/http"
```

```
#recebendo os dados via post
parametros = {'chaveAPI' => params[:chave], 'usuarioNome' => params[:usuario], 'numeroTel'
=> params[:tel], 'mensTexto' => params[:msg]}
```

```
#executando a acao de post dos parametros no servidor de sms
x = Net::HTTP.post_form(URI.parse(https://smsbr.com.br/enviosms.php), parametros)
```

### ***HTTP POST – Utilizando sms com Delphi***

Aplicação simples , utilizando uma `StringList` e o componente `idHttp` da paleta **Indy**:  
procedure TForm1.Button1Click(Sender: TObject);

var

D: TStringList;

begin

D := TStringList.Create; try

D.Add('chaveAPI=' + Edit2.Text);

D.Add('usuarioNome=' + Edit1.Text);

D.Add('numeroTel=' + Edit3.Text);

D.Add('mensTexto=' + memo1.Text);

D.Add('assinatura=' + Edit3.Text);

try

Memo2.Lines.Add (idHTTP1.Post('https://smsbr.com.br/enviosms.php',D));

except

MessageBox(0,'Ocorreu um erro durante o envio. Tente novamente','Erro',MB\_ICONERROR);

end;

finally

MessageBox(0,'Informações enviadas com sucesso','Informação',MB\_ICONINFORMATION);

end;

end;

Download do exemplo acima:

[http://www.4shared.com/rar/aKciuS8V/Simples\\_Post\\_SMSBR.html](http://www.4shared.com/rar/aKciuS8V/Simples_Post_SMSBR.html)

### 3. Retorno de Envio

Após o envio nossos servidores irão lhe retornar um status inicial de entrega.

Código status

| STATUS                | DESCRIÇÃO  |
|-----------------------|--|
| ERROR_AUTENTICACAO    | Erro de autenticação, chave api ou login incorretos  |
| ERROR_USER_SUPENSO    | Usuário suspenso, provavel bloqueio por uso indevido   |
| ERROR_SALDO           | Saldo em conta insuficiente para o envio das mensagens   |
| ERROR_DDI             | DDI informado inválido   |
| ERROR_LIMIT_CHARACTER | Limite de caracteres da mensagem superior a 160  |
| ERROR_MSGID_EM_USO    | ID_PROPIO informado já foi utilizado   |
| ERROR_MSG_VAZIO       | Campo mensagem não informado   |
| ALERTA_PHISHING       | Mensagem contém palavras que comumente são utilizadas em mensagens mal intencionadas, usuário é bloqueado para averiguação do setor responsável. |

### 4. Consultar saldo

A URL de destino para a instrução dos dados:

[https://smsbr.com.br/saldo.php?chaveAPI=CHAVE\\_API&usuarioNome=SEU\\_LOGIN](https://smsbr.com.br/saldo.php?chaveAPI=CHAVE_API&usuarioNome=SEU_LOGIN)

### 5. Retorno automático de STATUS

Além da comunicação através de webservices, é possível também automatizar a atualização de status e recebimento de Mos, através dos métodos HTTP

GET, HTTP POST

Nesse caso, assim que acontece a atualização do status para uma mensagem enviada pelo cliente, o sistema envia para uma URL pré estabelecida, através do método GET ou POST, para um endereço, o mesmo acontece para o recebimento de mensagens de resposta ( Mos ) .

Para os Mos o caminho de retorno permanece ativo durante um período de 4 dias

Após esse período, as mensagens de resposta são descartadas.

O servidor deve ser configurado para receber os eventos onde serão passados os parâmetros necessários.

O formato da URL, para o método GET é a seguinte:

Exemplo:

<https://urldocliente.com.br/?envioData=2019-12-19T01:35:59&APIMsgID=EV018TW5&status=001&celular=5584944556655>

Parâmetros de STATUS:

- envioData (data do envio)
- APIMsgID (ID da Mensagem)
- celular (número do destinatário)
- status (3 números, veja abaixo\*)

### **Legenda dos status**

- 001: Mensagem desconhecida
- 002: Mensagem na fila de entrega
- 003: Entregue ao Gateway
- 004: Recebida pelo celular
- 005: Erro na mensagem
- 006: Cancelada pelo usuário
- 007: Erro na entrega da mensagem
- 008: Recebida pelo gateway
- 009: Erro no encaminhamento
- 010: Mensagem expirou
- 011: Enfileiradas para posterior entrega
- 012: Numero inválido
- 013: Numero de origem inválido
- 014: Rota não disponível