Integração Web Services
REST Gateway

Índice de Conteúdos

Introdução Page 3.
Plataforma técnica
Solicitação de envios SMS Pág. 3.
Solicitação JSON Pág. 3
Exemplo solicitação CURL Pág. 4
Exemplo de solicitação PHP Pág. 5
Códigos do status de resposta Pág. 6
Apêndice A: Aviso de recepção Pág. 7.
Exemplo de solicitação CURL Pág. 8
Exemplo de solicitação PHP Pág. 9
Apêndice B: Conjunto de caracteres GSM7 Pág. 10.

Introdução

A Plataforma REST Gateway permite ao usuário enviar mensagens através de HTTP ou HTTPS de forma simples e rápida, podendo enviar mais de 500 mensagens em uma única solicitação. Para poder ter acesso a suas estatísticas e dados de faturamento, pode aceder a página web http://plusmms.net com os seus dados de usuário.

Esta documentação descreve os parâmetros necessários e facultativos para a utilização de todas as possibilidades no envio de mensagens sms seguindo as especificações REST. Tanto as solicitações como a as respostas da API REST estão em formato JSON, tornando fácil a utilização da API com qualquer linguagem de programação.

PLATAFORMA TÉCNICA

Solicitação de envio de SMS

Cada solicitação que se realize terá que incluir no cabeçalho da solicitação http a autenticação do cliente. Para tal se utiliza a autenticação de acesso básico do HTTP.

O cabeçalho de autorização se constrói combinando a sequência "usuário: contrasenha" e codificando-a em base64. A esta sequência se antepõe a sequência "Autorização: Basic"

Por exemplo, para o usuário "miuser" e a contra-senha "mipass" o cabeçalho resultante seria:

Authorization: Basic bWl1c2VyOm1pcGFzcw==

A seguir, se detalharão as opções de envio disponíveis, o URL ao qual se deverá chamar e os parâmetros admitidos.

Para gerar a URL o cliente deverá fazer uma chamada POST ao seguinte endereço:

https://gateway.plusmms.net/rest/message

Solicitação JSON:

Exemplo de solicitação básica:

{"to":["34666555444"],"text":"mensaje de texto","from":"msg"}

Possíveis parâmetros:

- **text:** texto da mensagem. Poderá conter no máximo 160 caracteres caso não se especifique que a mensagem é multiparte (veja parâmetro "parts"). O texto deve ser codificados em UTF-8
- **to:** número do celular destinatário da mensagem. Deve incluir o prefixo (p. ex: No Brasil 55666666666). Este campo permite que se especifiquem

vários destinatários, para tal deve incluir todos os números em uma matriz.

- **from**:Texto do Remetente, esta etiqueta será composta por 15 números ou 11 caracteres alfanuméricos.
- **coding (opcional):** Os valores possíveis são "gsm" e "utf-16". O valor "gsm" para envios normais com codificação GSM7 e 160 caracteres por mensagem e o valor "utf-16" para a codificação UCS2 (UTF16) e 70 caracteres por mensagem. Se não estiver especificado, o valor padrão será "gsm"
- **fSend(opcional):** Data no envio da mensagem. Caso seja necessário enviar mensagens programadas se pode especificar a data de envio indicando a data no formato YYYYmmddHHiiss (p. ex: 20130215142000 seria 15 de fevereiro de 2013 às 14:20). No caso de envio imediato não é necessário especificar este parâmetro.
- **parts(opcional):** Indica o número máximo de partes em que se dividirá a mensagem ao ser enviada. Esta variável é definida por padrão com valor 1, portanto se não for especificado, ao enviar uma mensagem com mais de 160 caracteres para codificação "GSM", a mensagem irá falhar. Tenha em mente que as mensagens concatenadas podem ter apenas 153 caracteres cada uma e cada uma das partes se tarifa como um envio. O servidor só utilizará o mínimo de partes necessárias para enviar as mensagens de texto, mesmo que o número especificado de partes seja maior do que o necessário. No caso em que o número de partes seja menor do que o necessário para enviar mensagens de texto, o envio falhará com o erro 105.
- **trsec(opcional)**: Tipo booleano. Com o valor "false" o servidor não modifica quaisquer caracteres na mensagem, este é o valor padrão. Com o valor "true" o servidor se encarrega de transformar os caracteres comuns inválidos no GSM7 em caracteres válidos com a seguinte tabela de tradução: 'á' => 'a', 'í'=>'i', 'ó'=>'o', 'ú'=>'u', 'ç'=>'Ç', 'Á'=>'A', 'Í'=>'I', 'Ó'=>'O', 'Ú'=>'U', 'À'=>'A', 'È'=>'E', 'Ì'=>'I', 'Ò'=>'O', 'Ù'=>'U', 'º' => '', 'ª' => '', 'Õ' => 'O', 'õ' => 'o', 'â' => 'a', 'ê' => 'e', 'î'=>'i', 'ô'=>'o', 'û'=>'u', 'Â'=>'A', 'Ê'=>'E', 'Î'=>'I', 'Ô'=>'O', 'Û'=>'U', 'ã' => 'a', 'Ã' => 'A'

Solicitação de amostra CURL:

```
curl -X POST \ -H "Content-Type: application/json" \ -H "Accept: application/json" \ -H "Authorization: Basic bWl1c2VyOm1pcGFzcw==" \ -d "{\"to\":[\"34666555444\"],\"text\":\"mensaje de texto\",\"from\":\"msg\"}" \ https://gateway.plusmms.net/rest/message
```

Exemplo de solicitação PHP:

```
<?php $post['to'] = array('34666555444'); $post['text'] = "mensaje de texto"; $post['from'] = "msg";  $user ="miuser"; $password = 'mipass';  $ch = curl_init(); curl_setopt($ch, CURLOPT_URL, "https://gateway.plusmms.net/rest/message"); curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1); curl_setopt($ch, CURLOPT_POST, 1); curl_setopt($ch, CURLOPT_HEADER, 1); curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($post)); curl_setopt($ch, CURLOPT_HTTPHEADER, array( "Accept: application/json", "Authorization: Basic ".base64_encode($user.":".$password) ));  $result = curl_exec ($ch); ?>
```

A senha (password) e o código do cliente (username) serão fornecidos pela empresa. Devemos mencionar que, a fim de melhorar a segurança do sistema, o cliente deverá indicar o IP de onde se conectará, somente serão permitidos envios provenientes do IP especificado pelo cliente.

Códigos de status de respostas

A API REST pode responder com os seguintes estados HTTP:

Código do estado	Descrição	Detalhes
202	Accepted	A mensagem foi aceita para seu processo posterior
207	Multi-status	A mensagem foi aceita para seu processo posterior, mas alguns destinatários estão incorretos.
400	Bad request	A solicitação contém erros, a mensagem não foi aceita
401	Unauthorized	Falha na autenticação do cliente
402	Payment required	O cliente não tem saldo suficiente
500	Internal server error	O servidor teve um erro interno

No corpo da resposta HTTP se entrega um JSON com os detalhes do resultado, estas são as possíveis respostas:

Código do estado 202:
[{"accepted":true,"to":"34666555444","id":"102648819"}]

Código do estado 207:
[{"accepted":true,"to":"34626690739","id":"102648820"},{"accepted":false,"to":"34","error":{"code":102,"description":"No valid recipients"}}]

Código do estado 400:
{"error":{"code":102,"description":"No valid recipients"}}
{"error":{"code":104,"description":"Text message missing"}}
{"error":{"code":105,"description":"Text message too long"}}
{"error":{"code":106,"description":"Sender missing"}}
{"error":{"code":107,"description":"Sender too long"}}
{"error":{"code":108,"description":"No valid Datetime for send"}}
{"error":{"code":109,"description":"Notification URL incorrect"}}
{"error":{"code":110,"description":"Exceeded maximum parts allowed or incorrect number of parts"}}
{"error":{"code":113,"description":"Invalid coding"}}

Código estado 401:
{"error":{"code":103,"description":"Username or password unknown"}}
{"error":{"code":112,"description":"IP address not allowed"}}

Código estado 402:
{"error":{"code":111,"description":"Not enough credits"}}

Apêndice A: Avisos de recebimento

Caso se deseje receber os avisos de recebimento em tempo real deverá ser especificada a variável "dlr-url" com o URL do cliente onde se deseja que o status do envio seja notificado.

O funcionamento consiste em especificar em cada solicitação HTTP o URL onde se deseja que seja realizada uma solicitação do nosso servidor ao receber uma notificação por parte da operadora. Para tal o cliente deve ter um servidor http capaz de receber tais notificações.

Nosso servidor enviará as variáveis pelo método GET da maneira que queira o cliente, por isso no URL que nos envie há que colocar o nome da variável seguido por um caractere de escape que conterá o valor, os caracteres de escape têm a forma do caractere "%", seguido de uma letra. Este seria um exemplo de URL:

http://mi.server.com/notifica.php?remitente=%p&tel=%P&estado=%d

Estes são os caracteres de escape definidos:

%i Identificador de NRS que foi entregue ao realizar o envio

%d valor do aviso de recebimento

%p o remetente da mensagem SMS

%P o número de telefone do destinatário do SMS

%t data do envio de mensagem com formato "YYYY-MM-DD HH:MM", por exemplo, "1999-09-21 14:18"

O valor %d é o que nos devolverá o estado final do envio, os valores possíveis são:

- 1: A mensagem foi entregue ao destinatário.
- 2: A mensagem não pôde ser entregue ao destinatário.
- 4: A mensagem foi entregue ao SMSC, é uma notificação intermediária, e não um resultado final
- 16: Não foi possível entregar a operadora final

Para explicar melhor o processo, a seguir, apresentamos um exemplo de como seria o envio de um sms e a recepção de seu aviso de recebimento.

Primeiro vamos enviar o sms com a variável dlr-url onde indicaremos a URL onde queremos receber o aviso de entrega, adicionaremos a esta URL nosso identificador de envio para identificá-lo inequivocamente ao recebê-lo. O URL final para a notificação seria: http://mi.server.com/notifica.php?idenvio=7584remitente=%p&tel=%P&estado=%d

Exemplo de solicitação CURL:

```
curl -X POST \ -H "Content-Type: application/json" \ -H "Accept: application/json" \ -H "Authorization: Basic bWl1c2VyOm1pcGFzcw==" \ -d "{\"to\":[\"34666555444\"],\"text\":\"mensaje \",\"from\":\"msg\",\"dlr-url\":\"http://mi.server.com/notifica.php?remitente=%p&tel=%P&estado=%d\"}" \ https://gateway.plusmms.net/rest/message
```

Exemplo de solicitação PHP:

```
<?php $post['to'] = array('34666555444'); $post['text'] = "mensaje de texto";
$post['from'] = "msg";
$post['dlr-url'] =
"http://mi.server.com/notifica.php?idenvio=7584remitente=%p&tel=%P&estado=%
d"  $user ="miuser"; $password = 'mipass';  $ch = curl_init(); curl_setopt($ch,
CURLOPT_URL,          "https://gateway.plusmms.net/rest/message");
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1); curl_setopt($ch,
CURLOPT_POST,         1); curl_setopt($ch, CURLOPT_HEADER,         1);
curl_setopt($ch, CURLOPT_POSTFIELDS,    json_encode($post)); curl_setopt($ch,
CURLOPT_HTTPHEADER,    array(    "Accept: application/json",    "Authorization:
Basic ".base64_encode($user.":".$password) ));  $result = curl_exec ($ch); ?>
```

Assumindo que todas as mensagens possam ser entregues, receberemos do script notifica.php três petições com o estado= 1, remetente=TEST, idenvio = 7584 e o número do telefone correspondente.

Conjunto de caracteres básicos

	0x00	0x10	0x20	0x30	0x40	0x50	0x60	0x70
0x00	@	Δ	SP	0	i	P	¿	p
0x01	£	_	!	1	A	Q	a	q
0x02	$	Φ	"	2	B	R	b	r
0x03	¥	Γ	#	3	C	S	c	s
0x04	è	Λ	¤	4	D	T	d	t
0x05	é	Ω	%	5	E	U	e	u
0x06	ù	Π	&	6	F	V	f	v
0x07	ì	Ψ	'	7	G	W	g	w
0x08	ò	Σ	(	8	H	X	h	x
0x09	Ç	Θ	)	9	I	Y	i	y
0x0A	LF	Ξ	*	:	J	Z	j	z
0x0B	Ø	ESC	+	;	K	Ä	k	ä
0x0C	ø	Æ	,	<	L	Ö	l	ö
0x0D	CR	æ	-	=	M	Ñ	m	ñ
0x0E	Å	ß	.	>	N	Ü	n	ü
0x0F	å	É	/	?	O	§	o	à

Extensão do conjunto de caracteres básicos, estes caracteres ocupam duas posições

	0x00	0x10	0x20	0x30	0x40	0x50	0x60	0x70
0x00					|			
0x01								
0x02								
0x03								
0x04		^						
0x05							€	
0x06								
0x07								
0x08			{					
0x09			}					
0x0A	FF							
0x0B		SS2						
0x0C				[				
0x0D	CR2			~				
0x0E				]				
0x0F			\					