interweb

Gerado a 13/11/2026 17:58

BASE DE CONHECIMENTO · DOCUMENTO PÚBLICO

Uma função para enviar SMS em PHP, Python ou Node.js, com tempo limite e tratamento de erros

WhatsApp e APIs

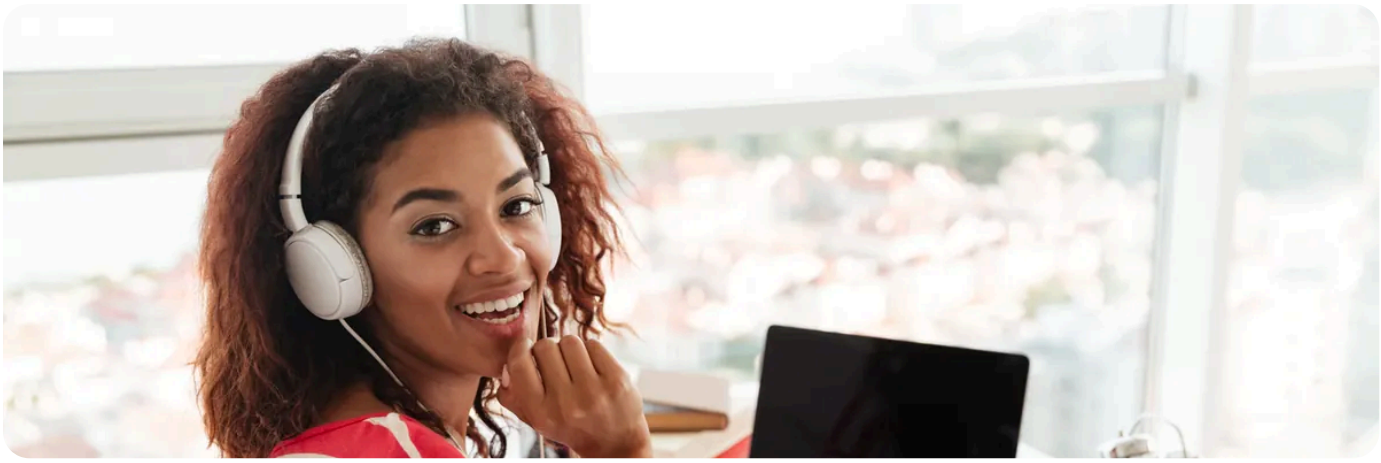
CÓDIGO	VERSÃO	PAÍS · IDIOMA
RLX-AO-KB-2602	v2026.11.13	Angola · Português

Interweb Angola
RLX Angola, Lda. · Grupo RLX Holdings
NIF 500 260 5724 · Matrícula 26464-25/250724

https://www.meu.interweb.ao/knowledgebase/2602

Uma função para enviar SMS em PHP, Python ou Node.js, com tempo limite e tratamento de erros

WhatsApp e APIs Criado em **13 de novembro de 2026** 3 min de leitura



A resposta curta: escreva **uma só função** que envia o SMS e use-a em todo o sistema. Ela recebe o número, o texto e uma chave do evento, lê a chave da API de uma variável de ambiente, tem **tempo limite**, envia sempre `Idempotency-Key` e devolve o código HTTP e a resposta. A decisão de repetir fica num só sítio: repete-se com espera num `429` ou `5xx`, nunca num `422`.

Em PHP

```
function enviar_sms($numero, $texto, $chave_evento) {
$ch = curl_init('https://api.interweb.ao/v1/sms/messages');
curl_setopt_array($ch, [
CURLOPT_RETURNTRANSFER => true,
CURLOPT_POST => true,
CURLOPT_TIMEOUT => 30,
CURLOPT_HTTPHEADER => [
'Authorization: Bearer ' . getenv('SMS_API_KEY'),
'Content-Type: application/json',
'Idempotency-Key: ' . $chave_evento,
],
CURLOPT_POSTFIELDS => json_encode(['number' => $numero, 'text' => $texto]),
]);
$corpo = curl_exec($ch);
$http = curl_getinfo($ch, CURLINFO_HTTP_CODE);
curl_close($ch);
return [$http, json_decode((string) $corpo, true)];
}
```

Em Python

```
import os, requests

def enviar_sms(numero, texto, chave_evento):
r = requests.post(
"https://api.interweb.ao/v1/sms/messages",
headers={"Authorization": "Bearer " + os.environ["SMS_API_KEY"],
"Idempotency-Key": chave_evento},
json={"number": numero, "text": texto},
timeout=30,
)
return r.status_code, r.json()
```

Em Node.js

```
async function enviarSms(numero, texto, chaveEvento) {
const r = await fetch("https://api.interweb.ao/v1/sms/messages", {
method: "POST",
headers: { "Authorization": "Bearer " + process.env.SMS_API_KEY,
"Content-Type": "application/json",
"Idempotency-Key": chaveEvento },
body: JSON.stringify({ number: numero, text: texto }),
signal: AbortSignal.timeout(30000),
});
return [r.status, await r.json()];
}
```

A `chave_evento` é o que torna a repetição segura: algo como `pagamento-3381` ou `encomenda-8812-saiu`. Com o mesmo valor e o mesmo corpo, a API devolve a resposta guardada em vez de enviar outra vez.

O que fazer com a resposta

Código	O que fazer
201	Guardar o `id` e o `status` junto do evento. Feito.
422	Não repetir. Ler `details.reason` (por exemplo `invalid_number` ou `insufficient_credits`) e corrigir ou avisar alguém da equipa.
401 e 403	Não repetir. A chave falta, está errada, expirou, não tem a permissão `sms:write`, o IP não está na lista dela, ou a conta não tem o SMS activo (`sms_reseller_required`).

Código	O que fazer
409	A mesma chave de evento com outro texto, ou o primeiro pedido ainda em curso. Esperar e repetir igual, ou usar uma chave nova se for mesmo outro envio.
429, 500, 502, 503	Repetir mais tarde, com a mesma chave de evento. Num 429, esperar pelo menos o tempo do cabeçalho `Retry-After`.
Sem resposta (tempo esgotado)	Repetir com a mesma chave de evento: se o primeiro pedido chegou, a API não envia segundo SMS.

Não envie o SMS dentro do pedido do cliente.
Se a página de pagamento ficar à espera da API de SMS, um atraso no SMS atrasa o pagamento. Ponha o envio numa fila ou numa tarefa à parte, e deixe a função repetir por lá.

Guarde o `correlation_id` dos erros
(vem na resposta e no cabeçalho `X-Correlation-ID`). Se precisar de ajuda, é o que o suporte pede, junto com a hora e a operação. A chave da API, essa, nunca se envia a ninguém.

RECOMENDADO

Pronto para ligar o seu sistema? Crie uma chave de Sandbox e comece pelo SMS Transaccional.

Ver o SMS Transaccional →

VEJA TAMBÉM

- **Enviar SMS pela API: chave, permissões, Sandbox e o primeiro pedido** →
- **Erros e estados das mensagens na API de SMS: o que significa cada um e o que fazer** →
- **Tempos de espera, repetições e idempotência: como não cobrar duas vezes ao cliente** →
- **Casos de teste do SMS transaccional na Sandbox: números errados, textos longos, saldo e repetições** →
- **Acompanhar cada SMS transaccional: o id da mensagem, o número tapado, os estados e o charged** →
- **Uma chave de API ou um token ficou exposto: o que fazer, pela ordem certa** →

interweb

Interweb Angola

RLX Angola, Lda. · Grupo RLX Holdings

NIF	500 260 5724
MATRÍCULA	26464-25/250724
SEDE	Rua Direita do Patriota, Bairro Dangereux Belas, Luanda, Angola
CONTACTOS	info@interweb.ao · +244 942 011 191 · www.interweb.ao
HORÁRIO	2ª. à 6ª. Feira - 08 à 16 Horas Sábado & Domingo - Fechado

Versão online deste documento

RLX-AO-KB-2602 · v2026.11.13 · Gerado a 13/11/2026 17:58