Mozout

Generated on 10/11/2026 09:39

KNOWLEDGE BASE · PUBLIC DOCUMENT

Verification codes by SMS through our API: a practical guide for your site or app

WhatsApp & APIs

CODE	VERSION	COUNTRY · LANGUAGE
RLX-INT-KB-7176	v2026.11.10	International · English

MozOut International
RLX Holdings (Pty) Ltd. · Group RLX Holdings
Tax no. (SARS) 943 907 6234 · Registration no. (CIPC) 2025/667953/07

https://www.my.mozout.com/knowledgebase/7176

Verification codes by SMS through our API: a practical guide for your site or app

WhatsApp & APIs Created on **November 10, 2026** 4 min read



The short answer: your site or app **creates the code, saves it and checks it**; our API **sends the SMS**. One `POST /v1/sms/messages` request with the number and the text is all it takes. Test in the **Sandbox** first, where nothing is sent or charged, then move to **Production** with a key that can only send SMS.

Who does what

Task	Who does it
Create a random code	Your system
Save the code (hashed, never in plain text)	Your system
Give the code a short expiry and limit attempts	Your system
Send the SMS to the phone	**Our API**
Check the code the person types in	Your system

The security side, in detail, is in **sending one-time codes safely**. This guide covers the practical part: the key, the request, the wording and what to do when something goes wrong.

Before the first code

1. **Activate SMS and buy credits** on the SMS page in My MozOut. A short code usually fits in one part, which is 1 credit.

2 **Create a Sandbox key** (it starts with `sandbox_`) on the API page in My MozOut, with the `sms:write` permission. We ask for your password and a 6-digit code sent by e-mail, and the key is **shown only once**: keep it on your server, never in a web page or a mobile app.

3 **Once everything works, create a Production key** (`prod_...`) with the same permission, a **spending limit** and, if your server has a fixed IP, a list of allowed IPs.

The request that sends the code

```
curl https://api.mozout.com/v1/sms/messages \
-H "Authorization: Bearer sandbox_..." \
-H "Content-Type: application/json" \
-H "Idempotency-Key: 7d4f2b1e-9c3a-4e8f-b6d2-1a5c3e9f0b7d" \
-d '{"number": "+258841234567", "text": "Your verification code is 482913."}'
```

In the Sandbox, the response comes back with `"status": "simulated"` and the `X-API-Simulated: true` header: the text was validated, but nothing was sent or charged. In Production, the same request sends the SMS and takes the credits from your SMS balance. The response includes the message `id`, the parts and the credits used; keep the `id` to check the status later with `GET /v1/sms/messages/{id}`.

The wording: short, in a single part

A code fits easily in a one-part SMS. In Portuguese, though, a single accent such as ó or ã switches the whole message to Unicode, and then one part holds only 70 characters. Keep code messages short and they stay in one part either way. The full count is in **how many credits an SMS uses**.

Always include three things: **who is sending** (your company name), **the code** and **a warning not to share it**. For example: `COMPANY: your code is 482913. Do not share it with anyone.`.

With a Sender ID, the SMS arrives under your company name
, so people recognise the code at once. Pass it in the `sender` field, using an approved name (see `GET /v1/sms/senders`). Without `sender`, the message goes out from the free NETMSG sender.

Double clicks, resends and limits

1 **Use the `Idempotency-Key` header.** Create a UUID for each code and reuse it if you have to repeat the request (a network error, a double click on “Send code”). The same key with the same body returns the same response and no second SMS; with a different body, you get a 409.

2 **Control the “Resend” button in your own system.** Make people wait a little before they can resend, and cap the number of resends per number: every resend is a new SMS and uses credits.

3 **Watch your balance and the limits.** The API accepts 300 requests per minute per account, and there is a daily sending limit per account. `GET /v1/sms/balance` returns your credit balance, `daily_limit` and `sent_today`.

! **A message that fails is tried twice more, 10 minutes apart.**
With a short expiry, the code may already be out of date by the time it goes out. So "Resend" should always create a fresh code. If a request is refused (for example, with `invalid_number`: "Check the digits, as in 841234567"), show the person a plain message and see what to do in **SMS API errors and statuses**.

RECOMMENDED

Want to send verification codes from your site or app? See what the API does and create your key.

See the API page →

SEE ALSO

- Sending SMS through the API: key, permissions, Sandbox and your first request →
- SMS API errors and message statuses: what each one means and what to do →
- Sending one-time codes (OTP) by SMS safely: expiry, attempt limits and logs →
- How many credits an SMS uses: 160, 153, 70 and 67 characters explained →
- What a Sender ID is: SMS that arrive with your company name instead of a number →
- Authenticator app, SMS or security key: which two-step verification to use →

Mozout

MozOut International

RLX Holdings (Pty) Ltd. · Group RLX Holdings

TAX NO. (SARS)	943 907 6234
REGISTRATION NO. (CIPC)	2025/667953/07
REGISTERED OFFICE	1111 Olive Street Saint Louis 63101, Missouri, United States
CONTACT	info@mozout.com · +27 78 823 3325 · www.mozout.com
OPENING HOURS	Monday to Friday - 08 AM to 04 PM Saturday & Sunday - Closed

Online version of this document

RLX-INT-KB-7176 · v2026.11.10 · Generated on 10/11/2026 09:39