Mozout

Generated on 13/11/2026 19:44

KNOWLEDGE BASE · PUBLIC DOCUMENT

# SMS verification API: we generate, send and check the code, and your system keeps only the id

WhatsApp & APIs

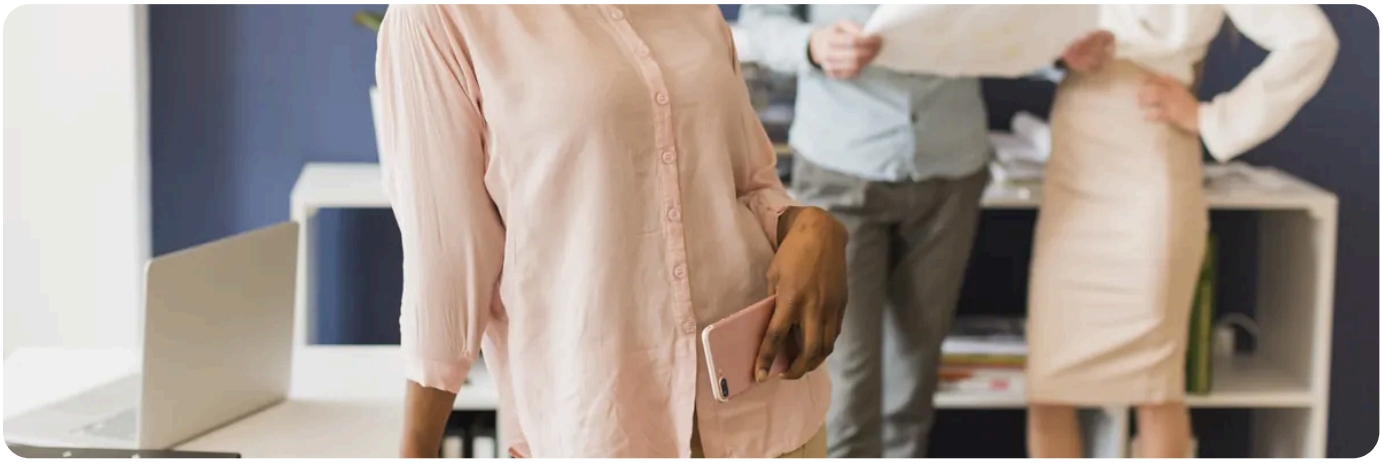
| CODE | VERSION | COUNTRY · LANGUAGE |
| --- | --- | --- |
| RLX-MZ-KB-2625 | v2026.11.13 | Mozambique · English |

**MozOut Mozambique**
RLX Holdings, S.A. · Group RLX Holdings
Tax ID (NUIT) 401 191 607 · Legal entity no. (NUEL) 105 045 974 · Economic activity code (CAE) 62022

https://www.meu.mozout.co.mz/knowledgebase/2625

# SMS verification API: we generate, send and check the code, and your system keeps only the id

WhatsApp & APIs Created on **November 13, 2026** 3 min read



**The short answer:** with our verification API, your system does not have to create, store or compare codes. It sends one `POST /v1/sms/otp` request with the customer's number; we create a random code, send it by SMS straight away and keep only its hash. When the customer types the code, your system calls `POST /v1/sms/otp/{id}/verify` and gets `valid`: yes or no. All you keep is the code's `id`, which starts with `otp_`.

## Two requests are enough

1. **The customer asks for a code.** On sign-up, login or payment, your system sends the number. Every other field is optional.

2. **We create it and send it.** The code goes out by SMS at once, ahead of your account's other messages. The response gives you the `id`, the `pending` status, the masked number and the time the code expires.

3. **The customer types the code** on your website or in your app. They never reply to the SMS: named senders cannot receive replies.

4. **We check it.** Your system sends what was typed and gets back `valid`, the status and the attempts left.

```
curl https://api.mozout.co.mz/v1/sms/otp \
-H "Authorization: Bearer sandbox_..." \
-H "Content-Type: application/json" \
-H "Idempotency-Key: signup-7f3a9c" \
-d '{"number": "841234567"}'
```

## The operations, one by one

| Operation | What it does | Permission |
|---|---|---|
| `POST /v1/sms/otp` | Create and send a code | `sms:write` |
| `POST /v1/sms/otp/{id}/verify` | Check the code the customer typed | `sms:write` |
| `POST /v1/sms/otp/{id}/resend` | Send a new code under the same `id` | `sms:write` |
| `POST /v1/sms/otp/{id}/cancel` | Make a pending code stop working | `sms:write` |
| `GET /v1/sms/otp/{id}` | See the status (never the code) | `sms:read` |
| `GET /v1/sms/otp-settings` | See your account's sending rules | `sms:read` |
| `PUT /v1/sms/otp-settings` | Change those rules | `sms:write` |

## What is already done, and what is still yours

| Done by the API | Still up to your system |
|---|---|
| **A random code of 4 to 8 digits (6 by default)** | Linking the `id` to the customer and to what they are doing (the session, the transaction) |
| **Validity from 1 to 60 minutes (5 by default)** | Deciding what happens when `valid` is `true` |
| **Counted attempts, 1 to 10 (5 by default)** | Protecting your form from bots (see **fraud through your code form**) |
| **Only the hash is stored, checked in constant time** | A clear, honest code entry screen |
| **A new code for the same number cancels the previous one** | Keeping the API key on your server, never in a web page or a mobile app |

**Start in the Sandbox.**
With a `sandbox_` key, no SMS is sent and nothing is charged, and the send response includes the code itself so you can test the check. See **testing verification in the Sandbox**.

**Before the first request, SMS must be active on your account**

and have credits. Otherwise the API answers `403` with `sms_reseller_required`. Activate the service and buy credits on the SMS page in Meu MozOut. Each code uses the credits for its SMS, usually one; checking it costs nothing.

RECOMMENDED

Want to confirm numbers and logins without creating or storing codes? See how Verification SMS works.

See Verification SMS →

SEE ALSO

- Your own codes or our verification API: which way to send SMS codes from your system
- Testing SMS verification in the Sandbox: the code comes back in the response
- Code length, validity and attempts: choosing length, expires_in and max_attempts
- Adding SMS verification to your site: PHP, Node.js and Python examples
- Sending SMS through the API: key, permissions, Sandbox and your first request
- Sending one-time codes (OTP) by SMS safely: expiry, attempt limits and logs

Mozout

## MozOut Mozambique

**RLX Holdings, S.A. · Group RLX Holdings**

| | |
|---|---|
| TAX ID (NUIT) | 401 191 607 |
| LEGAL ENTITY NO. (NUEL) | 105 045 974 |
| ECONOMIC ACTIVITY CODE (CAE) | 62022 |
| REGISTERED OFFICE | Bairro 7 de Setembro<br>Vilankulo 1304 · CEP 0411-02, Inhambane, Mozambique |
| CONTACT | info@mozout.co.mz · +258 84 635 1111 · +258 86 635 1111 · www.mozout.co.mz |
| OPENING HOURS | Monday to Friday - 08 AM to 04 PM<br>Saturday & Sunday - Closed |

Online version of this document

RLX-MZ-KB-2625 · v2026.11.13 · Generated on 13/11/2026 19:44