interweb

Generated on 13/11/2026 18:18

KNOWLEDGE BASE · PUBLIC DOCUMENT

Idempotency-Key in our API: repeating a purchase, payment or ticket without doing it twice

WhatsApp & APIs

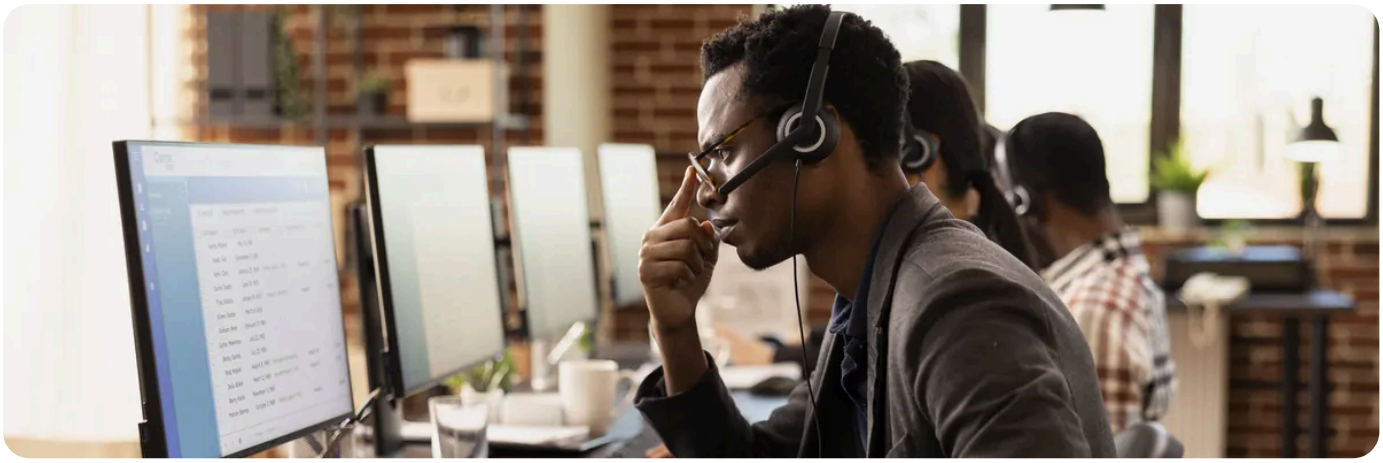
CODE	VERSION	COUNTRY · LANGUAGE
RLX-AO-KB-2764	v2026.11.13	Angola · English

Interweb Angola
RLX Angola, Lda. · Group RLX Holdings
Tax ID (NIF) 500 260 5724 · Commercial registration no. 26464-25/250724

https://www.meu.interweb.ao/knowledgebase/2764

Idempotency-Key in our API: repeating a purchase, payment or ticket without doing it twice

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



The short answer: if the connection drops in the middle of a request, you cannot tell whether it was carried out. With the `Idempotency-Key` header you can repeat the same request **without any risk of registering, charging or opening twice**: for **24 hours**, the same value with the same body returns the stored answer and nothing is done again.

How to use it

Generate a unique value for each operation, such as a UUID, and send it with a `POST` request. It can be up to 255 visible characters. Reuse the same value **only** when you repeat the same operation.

```
curl https://api.interweb.ao/v1/invoices/3127/pay -X POST \
-H "Authorization: Bearer $RLX_API_KEY" \
-H "Idempotency-Key: 0f8fad5b-d9cb-469f-a165-70867728950e"
```

What happens in each case

Situation	Answer
First request with this value	The operation runs normally.
Same value and same body	The stored answer comes back, with the `Idempotent-Replayed: true` header. Nothing is done again.
Same value and a different body	`409 idempotency_conflict`
Same value while the first request is still running	`409 idempotency_conflict`

The value stays reserved **per key** for 24 hours.

Where it is required

Operation	What it prevents
`POST /v1/domains/register`, `/renew`, `/transfer`	Registering, renewing or transferring the same domain twice
`POST /v1/extras/purchase`	Buying the same certificate or SiteLock twice
`POST /v1/invoices/{id}/pay`	Charging the same invoice twice
`POST /v1/tickets`	Opening the same ticket twice
`POST /v1/servers/{id}/reinstall`, `/restore`	Reinstalling or restoring the server twice
`POST /v1/sms/messages`	Sending the same SMS twice

For ticket replies, `POST /v1/tickets/{id}/replies`, it is recommended, so that a retry does not post the reply twice.

After a failure

1. **No answer arrived** (timeout, dropped connection): repeat the request with the **same** value.
2. **You got a `409` because the first is still running:** wait a moment and retry with the same value.
3. **You got a `409` because the body differs:** if it really is a different operation, use a new value.

Where the key does not protect you

> **Password requests and cPanel login:**
> For mailbox, database and FTP passwords, and for the cPanel login link, the `Idempotency-Key` **has no effect**: repeating the request runs it again, because the API never stores the password or the link.

Special case	What happens
VPS root password	Retrying with the same key does not generate another password and returns `password` as `null`. The password only appears in the first answer: save it straight away.
Hosting with a `502`	If the request had already reached the server, the same key returns that `502` without running it again. Check the result and, to try again, use a new key.

Generate the key before the first attempt and store it with the operation
(for example, on the order row in your database). That way, even if your program restarts halfway, the retry uses the same value. The general idea, for any API, is in **timeouts, retries and idempotency**.

RECOMMENDED

Is your program going to buy or pay through the API? Test your retries in Sandbox before you use a Production key.

See the API page →

SEE ALSO

- **Timeouts, retries and idempotency: how not to charge a customer twice** →
- **Our API's error codes: what each one means, and which ones are worth retrying** →
- **Long-running API operations: the 202 answer, the action_id and following the result** →
- **Domains through the API: checking a name, registering, renewing and transferring with credit** →
- **Invoices through the API: listing, PDF, paying with credit, renewal cycle and cancellations** →
- **Synchronous or asynchronous: why a payment or SMS confirmation arrives later** →

interweb

Interweb Angola

RLX Angola, Lda. · Group RLX Holdings

TAX ID (NIF)	500 260 5724
COMMERCIAL REGISTRATION NO.	26464-25/250724
REGISTERED OFFICE	Rua Direita do Patriota, Bairro Dangereux Belas, Luanda, Angola
CONTACT	info@interweb.ao · +244 942 011 191 · www.interweb.ao
OPENING HOURS	Monday to Friday - 08 AM to 04 PM Saturday & Sunday - Closed

Online version of this document

RLX-AO-KB-2764 · v2026.11.13 · Generated on 13/11/2026 18:18