interweb

Generated on 28/11/2026 22:01

KNOWLEDGE BASE · PUBLIC DOCUMENT

The request limit per key: 429 rate_limited and the Retry-After header

Using the AI API

CODE	VERSION	COUNTRY · LANGUAGE
RLX-MZ-KB-3010	v2026.11.28	Mozambique · English

Interweb 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.interweb.co.mz/knowledgebase/3010

The request limit per key: 429 rate_limited and the Retry-After header

Using the AI API Created on **November 28, 2026** 3 min read



In short: each key has a limit of requests per minute. When you go over it, the API answers `429` with the code `rate_limited` and the `Retry-After` header, which says how many seconds to wait. Your program only has to wait that long and try again.

The limits that count

Who is asking	Limit
Each key	A number of requests per minute, given in the `X-RateLimit-Limit` header.
The whole account, adding up its keys	300 requests per minute (1,000 for reseller accounts, which also have 300 per key).

These are the current values and they can change. In your program, read the headers instead of hard-coding the numbers.

Seeing the limit and what is left

Every response has three headers:

Header	What it is
`X-RateLimit-Limit`	Requests allowed per minute.
`X-RateLimit-Remaining`	Requests left in this minute.
`X-RateLimit-Reset`	When the minute ends, in Unix seconds.

To see them, add `-i` to curl. Listing the models spends no credits:

```
curl -i "https://api.interweb.co.mz/v1/models?type=text" \
  -H "Authorization: Bearer $INTERWEB_API_KEY"
```

`GET /v1/limits` shows the same as JSON, with the limit and the requests left in the minute:

```
curl "https://api.interweb.co.mz/v1/limits" -H "Authorization: Bearer $INTERWEB_API_KEY"
```

When you reach the limit

```
{
  "error": {
    "code": "rate_limited",
    "message": "Too many requests. Please try again shortly.",
    "correlation_id": "0f8fad5b-d9cb-469f-a165-70867728950e",
    "details": {"retry_after": 42}
  }
}
```

1. Wait at least the seconds in `Retry-After` (they also come in `details.retry_after`).
2. If the `429` comes back, wait longer on each attempt.
3. Avoid very fast loops, such as checking a generation several times a second.

An example in Python, which waits the time asked for and doubles the wait if the error comes back:

```
import os, time, requests

URL = "https://api.interweb.co.mz/v1/chat/completions"
HEADERS = {"Authorization": "Bearer " + os.environ["INTERWEB_API_KEY"]}

def send(body, attempts=5):
    wait = 1
    for _ in range(attempts):
        r = requests.post(URL, headers=HEADERS, json=body, timeout=60)
        if r.status_code != 429 or r.json()["error"]["code"] != "rate_limited":
            return r
        wait = max(int(r.headers.get("Retry-After", wait)), wait)
        time.sleep(wait)
        wait *= 2
    return r
```

The code checks for `rate_limited` because there is another `429`: `plan_limit_reached`, when the Coding Plan points run out. That one does not pass in seconds: see **plans and credits**.

What uses up requests without you noticing

Each status check of a generation counts as a request: check every 5 seconds or more. The MCP server tools, the `aio` commands and the requests of coding agents, which make many in a row, also count. The OpenAI SDKs retry a `429` on their own, twice by default. If you have several programs, give each its own key: the limit is per key, but the account limit adds them all up.

The limits of other API operations, such as domains and SMS, are in **API request limits**.

RECOMMENDED

Want to know what each key has spent? See the History in All in One AI.

See All in One AI →

SEE ALSO

- **API request limits: per key, per account and per task, the headers and what to do after a 429**
- **Error 429 Too Many Requests: who is asking too much, and what to do**
- **Creating many AI images at once: the balance, the price and the request limit**
- **AI API errors: 402, 403, 404, 422, 429 and 502, and what to do in each case**
- **All in One AI**

interweb

Interweb 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 Vilankulo 1304 · CEP 0411-02, Inhambane, Mozambique
CONTACT	info@interweb.co.mz · +258 84 635 1111 · +258 86 635 1111 · www.interweb.co.mz
OPENING HOURS	Monday to Friday - 08 AM to 04 PM Saturday & Sunday - Closed

Online version of this document

RLX-MZ-KB-3010 · v2026.11.28 · Generated on 28/11/2026 22:01