Mozout

Generated on 28/11/2026 22:36

KNOWLEDGE BASE · PUBLIC DOCUMENT

Errors in the OpenAI SDK: reading error.message and the HTTP status

Using the AI API

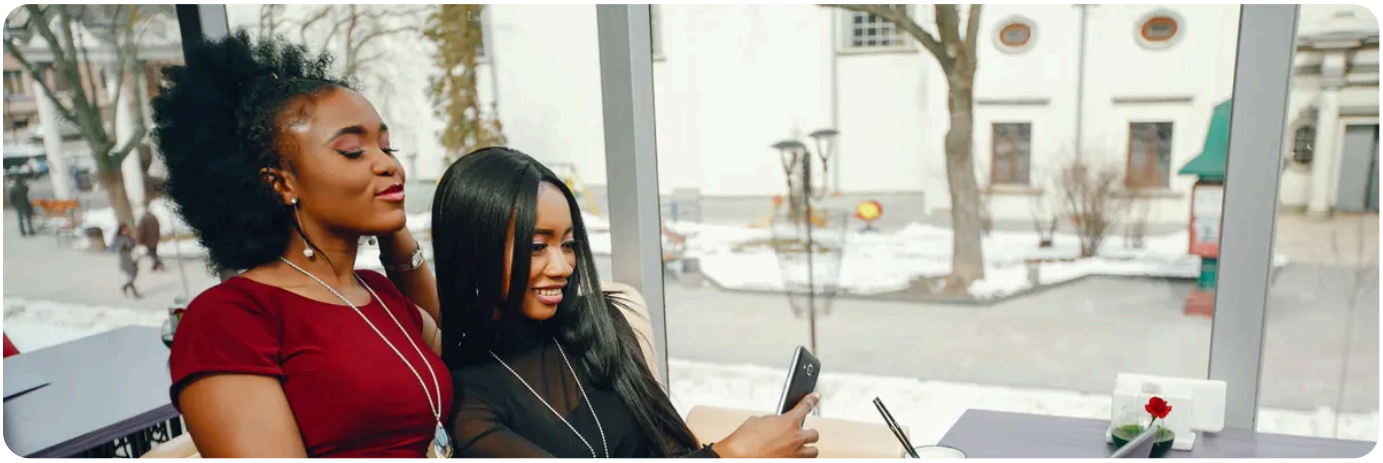
CODE	VERSION	COUNTRY · LANGUAGE
RLX-MZ-KB-3011	v2026.11.28	Mozambique · English

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/3011

Errors in the OpenAI SDK: reading error.message and the HTTP status

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



In short: when a request fails, the OpenAI SDK reads the message from `error.message` and raises an exception with the HTTP status. In Python it is an `openai.APIStatusError`, with `status_code`; in Node.js, an `OpenAI.APIError`, with `status`. The stable code, such as `insufficient_credits`, is in the response body, inside `error`.

In Python

```
import os
import openai
from openai import OpenAI

client                 =                 OpenAI(base_url="https://api.mozout.co.mz/v1",
api_key=os.environ["MOZOUT_API_KEY"])

try:
    completion = client.chat.completions.create(
        model="deepseek-ai/deepseek-v4-flash",
        messages=[{"role": "user", "content": "Hello"}],
        max_tokens=200,
    )
    print(completion.choices[0].message.content)
except openai.APIStatusError as e:
    error = e.response.json()["error"]
    print(e.status_code, error["code"], error["message"], error["correlation_id"])
except openai.APIConnectionError:
    print("Could not reach the API")
```

`e.response` is the whole HTTP response: from it you take the `code`, the message and the `correlation_id`. `APIConnectionError` happens when there was no response at all, for example on a network failure.

In Node.js

```
import OpenAI from "openai";

const client = new OpenAI({ baseURL: "https://api.mozout.co.mz/v1", apiKey: process.env.MOZOUT_API_KEY });

try {
  const completion = await client.chat.completions.create({
    model: "deepseek-ai/deepseek-v4-flash",
    messages: [{ role: "user", content: "Hello" }],
    max_tokens: 200,
  });
  console.log(completion.choices[0].message.content);
} catch (err) {
  if (err instanceof OpenAI.APIError) {
    console.log(err.status, err.message);
  } else {
    throw err;
  }
}
```

The `correlation_id` also comes in the `X-Correlation-ID` header of the response, which the error keeps in `err.headers`.

Which class goes with each status

The SDKs have one class for each status. You can catch just one, for example `openai.RateLimitError`, instead of all of them:

Status	SDK class	Our codes
400	`BadRequestError`	`invalid_request`
401	`AuthenticationError`	`unauthorized`
402	No class of its own: `APIStatusError` (Python) or `APIError` (Node.js)	`insufficient_credits`
403	`PermissionDeniedError`	`insufficient_scope`, `ip_not_allowed`, `account_*`
404	`NotFoundError`	`model_not_found`, `not_found`
422	`UnprocessableEntityError`	`validation_failed`

Status	SDK class	Our codes
`429`	`RateLimitError`	`rate_limited`, `plan_limit_reached`
500 or more	`InternalServerError`	`internal_error`, `upstream_error`, `disabled`

Automatic retries

The SDKs retry some errors on their own, twice by default and with a short wait: connection failures, `408`, `409`, `429` and `500` or more. You change it with `max_retries` in Python and `maxRetries` in Node.js; `0` turns it off. A `402`, `403`, `404` or `422` is not retried: fix the request first. And not everything that is retried goes through: a `429 plan_limit_reached` or a `409 agent_paused` stays the same until something changes.

In your program, decide by the `code` and not by the message: the message comes in the language of the `Accept-Language` header and can change. Keep the `correlation_id` in your program's log, to give it to support if you need to. The list of codes is in **the AI API errors**.

RECOMMENDED

Want to test your error handling without spending? Use a sandbox_ key and see the models in All in One AI.

See All in One AI →

SEE ALSO

- **AI API errors: 402, 403, 404, 422, 429 and 502, and what to do in each case**
- **Using the OpenAI SDK with All in One AI: only the address and the key change**
- **Our API's error codes: what each one means, and which ones are worth retrying**
- **The request limit per key: 429 rate_limited and the Retry-After header**
- **All in One AI**

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 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 Saturday & Sunday - Closed

Online version of this document

RLX-MZ-KB-3011 · v2026.11.28 · Generated on 28/11/2026 22:36