Mozout

Generated on 28/11/2026 22:35

KNOWLEDGE BASE · PUBLIC DOCUMENT

# The OpenAI Responses format: /v1/responses and responses kept for 30 days

Using the AI API

| CODE | VERSION | COUNTRY · LANGUAGE |
| --- | --- | --- |
| RLX-MZ-KB-3001 | 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/3001

# The OpenAI Responses format: /v1/responses and responses kept for 30 days

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



**In short:** besides OpenAI chat, the API accepts the OpenAI Responses format, at `POST /v1/responses`. It is what Codex CLI uses, and also `client.responses.create` in the OpenAI SDKs. It works like chat: the same models and agents, the same reservation and charge (or the Coding Plan points) and the same Sandbox. The difference that matters most: each response is kept for 30 days, so you can go on with the conversation without sending it again.

## The request

The base address ends in `/v1`, as in chat. The key needs `ai:write`:

```
curl "https://api.mozout.co.mz/v1/responses" \
  -H "Authorization: Bearer $MOZOUT_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{"model": "moonshotai/kimi-k2.7-code", "instructions": "You are a careful senior developer.", "input": "Write a Python function that checks if a string is a palindrome.", "max_output_tokens": 1024}'
```

| Field | What it is |
|---|---|
| `model` | The id of a text model from `GET /v1/models`, or `agent/` followed by the id of one of your agents. Required. |
| `input` | The text of the request, or a list of items (messages and function results). Required. |
| `instructions` | The instructions, the same as the system message. |
| `max_output_tokens` | The most tokens the answer may have. Without it, we reserve 4096. |
| `temperature` | From 0 to 2: lower is more predictable. |

| Field | What it is |
|---|---|
| `previous_response_id` | The `id` of an earlier response of the same account, to go on with that conversation. |
| `stream` | With `true`, the answer comes as Responses events, from `response.created` to `response.completed`. |

The answer has the `id` (it starts with `resp_`), the `status`, the text inside `output` (in a `message` item, with an `output_text` block) and the tokens in `usage`.

## Going on with previous_response_id

For the next question, you do not need to send the whole conversation again. Send the `id` of the earlier response in `previous_response_id` and only the new question in `input`:

```
{
  "model": "moonshotai/kimi-k2.7-code",
  "previous_response_id": "resp_7d2c9e4b1a6f3c8e5b0d2a19",
  "input": "Now write two tests for that function.",
  "max_output_tokens": 1024
}
```

It only works with responses of the same account from the last 30 days.

## Reading a saved response

`GET /v1/responses/{id}` returns the response as it was given. It needs `ai:read` and only the account that made it can see it:

```
curl "https://api.mozout.co.mz/v1/responses/resp_7d2c9e4b1a6f3c8e5b0d2a19" \
  -H "Authorization: Bearer $MOZOUT_API_KEY"
```

After 30 days, the answer is `404`. If you need the text for longer, keep a copy in your own system.

## Functions and errors

Function calls come as `function_call` items, each with its own `call_id`. Your program runs the function and answers in `input` with a `function_call_output` item, with the same `call_id` and the result in `output`. Errors come in the OpenAI format (`error.message`, `error.type`, `error.param` and `error.code`), with the same `code` as the rest of the API.

With a `sandbox_` key, the request answers in the same format without calling the model and without spending. For Codex CLI, the configuration file is in **Codex CLI with All in One AI**.

RECOMMENDED

Want to see the models you can use with Responses? They are in All in One AI.

See All in One AI →

SEE ALSO

- Codex CLI with All in One AI: creating config.toml, the key and the model
- Sending tools (function calling) in a chat request
- Using the OpenAI SDK with All in One AI: only the address and the key change
- Apps made for Claude: the Anthropic Messages format at /v1/messages
- All in One AI

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-3001 · v2026.11.28 · Generated on 28/11/2026 22:35