Mozout

Generated on 27/11/2026 08:07

KNOWLEDGE BASE · PUBLIC DOCUMENT

Showing the AI reply as it is written in your website or app (streaming)

All in One AI

CODE	VERSION	COUNTRY · LANGUAGE
RLX-INT-KB-8724	v2026.11.27	International · English

MozOut International
RLX Holdings (Pty) Ltd. · Group RLX Holdings
Tax no. (SARS) 943 907 6234 · Registration no. (CIPC) 2025/667953/07

https://www.my.mozout.com/knowledgebase/8724

Showing the AI reply as it is written in your website or app (streaming)

All in One AI Created on **November 26, 2026** Updated on **November 27, 2026** 2 min read



In short: send `stream: true` in your chat request and the reply arrives in pieces, as the model writes it. That is how text appears gradually in a chat window, instead of the person staring at a blank screen. Always add `stream_options` with `include_usage`.

What changes in the request

The request goes to the same address, `POST https://api.mozout.com/v1/chat/completions`, with the same key. You only add two fields. In Python, with the OpenAI SDK:

```
import os
from openai import OpenAI

client = OpenAI(base_url="https://api.mozout.com/v1", api_key=os.environ["MOZOUT_API_KEY"])
stream = client.chat.completions.create(
model="deepseek-ai/deepseek-v4-flash",
messages=[{"role": "user", "content": "Explain DNS in three short paragraphs."}],
max_tokens=600,
stream=True,
stream_options={"include_usage": True},
)
for chunk in stream:
if chunk.choices and chunk.choices[0].delta.content:
print(chunk.choices[0].delta.content, end="", flush=True)
```

The reply arrives as `text/event-stream` events and ends with `data: [DONE]`. Node.js and curl work the same way: see the examples under **With streaming** in the documentation.

Why include_usage matters

When streaming, the token count comes in the last event. If the model does not send it, we charge the whole reservation instead of what was used. With `include_usage`, you only pay for the real tokens. See **how a chat request is charged**.

Keep the key on your server.
Do not call the API straight from the browser or from a phone app: anyone who opens the code sees the key and spends your credits. The browser talks to your server, and your server talks to us and passes the pieces on to the browser. If a key has been exposed, see **what to do, in the right order**.

Before you go live

1. Write and test the code with a `sandbox_` key. The reply is always "Resposta de teste (sandbox).", streaming included, and costs nothing.
2. Set `max_tokens` so the reply cannot grow without limit.
3. Switch to the `prod_` key, kept in an environment variable, and watch the requests in your All in One AI History.

Not used the SDK with us yet? See **pointing the OpenAI SDK at our API**. Only the address and the key change.

RECOMMENDED

Want the Python, Node.js and curl examples? They are in the AI documentation.

Read the documentation →

SEE ALSO

- **What a chat request costs through the API: the reservation, max_tokens and the final adjustment**
- **Using the OpenAI SDK with All in One AI: only the address and the key change**
- **Testing All in One AI in the Sandbox at no cost: what each answer looks like**
- **An API key or token has leaked: what to do, in the right order**

Chat and text in All in One AI

Mozout

MozOut International

RLX Holdings (Pty) Ltd. · Group RLX Holdings

TAX NO. (SARS)	943 907 6234
REGISTRATION NO. (CIPC)	2025/667953/07
REGISTERED OFFICE	1111 Olive Street Saint Louis 63101, Missouri, United States
CONTACT	info@mozout.com · +27 78 823 3325 · www.mozout.com
OPENING HOURS	Monday to Friday - 08 AM to 04 PM Saturday & Sunday - Closed

RLX-INT-KB-8724 · v2026.11.27 · Generated on 27/11/2026 08:07

Online version of this document