Mozout

Generated on 28/11/2026 21:11

KNOWLEDGE BASE · PUBLIC DOCUMENT

An agent's answer with streaming: why the first piece can take longer

Agents Through the API

CODE	VERSION	COUNTRY · LANGUAGE
RLX-INT-KB-9246	v2026.11.28	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/9246

An agent's answer with streaming: why the first piece can take longer

Agents Through the API Created on **November 28, 2026** 2 min read



With `stream: true`, the answer arrives in pieces, as the model writes. In a chat, the text appears little by little and the person asking is not left staring at a still screen. With an agent it works the same way, with one difference: the first piece may take longer.

Why it takes longer

Before writing the final answer, the agent may use tools: check the date, read a page, search the web. Each use is a round to the model, and there can be up to 5 per request. Those rounds do not come through the stream. Only the final answer arrives in pieces, so while the agent works, the request stays open and nothing arrives.

A page read with `web_fetch` can take up to 10 seconds. An agent with no tools turned on starts writing sooner.

An example in Python

With the OpenAI SDK and the key in the `MOZOUT_API_KEY` variable:

```
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="agent/7c1e9a42-b35d8f06-a2c4e719-5d0b3f8e",
    messages=[{"role": "user", "content": "Are you open today?"}],
    max_tokens=400,
    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)
    if chunk.usage:
        print("\nTokens:", chunk.usage.total_tokens)
```

The stream ends with `data: [DONE]`. The SDK handles that: the loop ends on its own.

Do not forget include_usage

The token count arrives in the last piece, if you ask for `stream_options` with `include_usage`. If it does not arrive, the whole credit reservation stays charged. Always ask for it.

In your chat

1. Show a waiting sign, such as "Typing...", from the moment you send until the first piece arrives.
2. Give the request time. If the agent has tools, do not cut it off after a few seconds.
3. Use `max_tokens` to cap the length of the answer. The request can lower the agent's maximum, never raise it.

With a `sandbox_` key, streaming works too: you get "Resposta de teste (sandbox).", with `usage` at zero. Use it to test your chat code without spending. Streaming in general is covered in **showing the AI reply as it is written**.

RECOMMENDED

Want an agent typing live in your chat? See All in One AI.

See All in One AI →

SEE ALSO

- **Showing the AI reply as it is written in your website or app (streaming)**
- **Using your AI agent through the API: agent/ in the model field**
- **AI agent tools: reading web pages and knowing today's date**
- **Agent limits: 5 tool rounds, and max_tokens and temperature that can only go down**

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-9246 · v2026.11.28 · Generated on 28/11/2026 21:11

Online version of this document