Gerado a 27/11/2026 05:07

BASE DE CONHECIMENTO · DOCUMENTO PÚBLICO

Mostrar a resposta da IA aos poucos no seu site ou aplicação (streaming)

All in One AI

CÓDIGO	VERSÃO	PAÍS · IDIOMA
RLX-BR-KB-8724	v2026.11.27	Brasil · Português (Brasil)

MozOut Brasil
RLX Holdings (Pty) Ltd. · Grupo RLX Holdings
N.º fiscal (SARS) 943 907 6234 · N.º de registo (CIPC) 2025/667953/07

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

Mostrar a resposta da IA aos poucos no seu site ou aplicação (streaming)

All in One AI Criado em **26 de novembro de 2026** Atualizado em **27 de novembro de 2026** 2 min de leitura



A resposta curta: envie `stream: true` no pedido de chat e a resposta chega por partes, à medida que o modelo escreve. É assim que o texto vai aparecendo num chat, em vez de a pessoa ficar a olhar para uma tela vazia. Junte sempre `stream_options` com `include_usage`.

O que muda no pedido

O pedido vai para o mesmo endereço, `POST https://api.mozout.com/v1/chat/completions`, com a mesma chave. Só se juntam dois campos. Em Python, com o SDK da OpenAI:

```
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)
```

A resposta chega como eventos `text/event-stream` e acaba em `data: [DONE]`. Em Node.js e com curl é igual: veja os exemplos em **Com streaming**, na documentação.

Porquê o include_usage

No streaming, a contagem dos tokens vem no último evento. Se o modelo não a enviar, cobramos a reserva inteira em vez do que foi usado. Com `include_usage`, paga só os tokens reais. Veja **como se cobra uma conversa**.

A chave fica no seu servidor.
Não chame a API diretamente do navegador nem de uma aplicação de celular: quem abrir o código vê a chave e gasta os seus créditos. O navegador fala com o seu servidor, e é o seu servidor que fala conosco e passa as partes ao navegador. Se uma chave ficou exposta, veja **o que fazer, pela ordem certa**.

Antes de pôr no ar

1. Escreva e teste o código com uma chave `sandbox_`. A resposta é sempre «Resposta de teste (sandbox).», também em streaming, e não gasta nada.
2. Defina `max_tokens` para a resposta não crescer sem fim.
3. Troque pela chave `prod_`, guardada numa variável de ambiente, e veja os pedidos no Histórico do All in One AI.

Ainda não usou o SDK conosco? Veja **como apontar o SDK da OpenAI para a nossa API**. Só mudam o endereço e a chave.

RECOMENDADO

Quer os exemplos em Python, Node.js e curl? Estão na documentação da IA.

Ver a documentação →

VEJA TAMBÉM

- **Quanto custa uma conversa pela API: a reserva, o max_tokens e o acerto no fim** →
- **Usar o SDK da OpenAI** →
- **Testar na Sandbox sem custo** →
- **Uma chave de API ou um token ficou exposto: o que fazer, pela ordem certa** →

Chat e texto no All in One AI

MozOut Brasil

RLX Holdings (Pty) Ltd. · Grupo RLX Holdings

N.º FISCAL (SARS)	943 907 6234
N.º DE REGISTO (CIPC)	2025/667953/07
SEDE	72 York Road, Kensington Joanesburgo 2094, Gauteng, África do Sul
CONTACTOS	info@mozout.com · +27 78 823 3325 · www.mozout.com
HORÁRIO	2ª. à 6ª. Feira - 08 à 16 Horas Sábado & Domingo - Fechado

Versão online deste documento

RLX-BR-KB-8724 · v2026.11.27 · Gerado a 27/11/2026 05:07