Mozout

Generated on 26/11/2026 10:24

KNOWLEDGE BASE · PUBLIC DOCUMENT

Using the OpenAI SDK with All in One AI: only the address and the key change

What You Can Automate

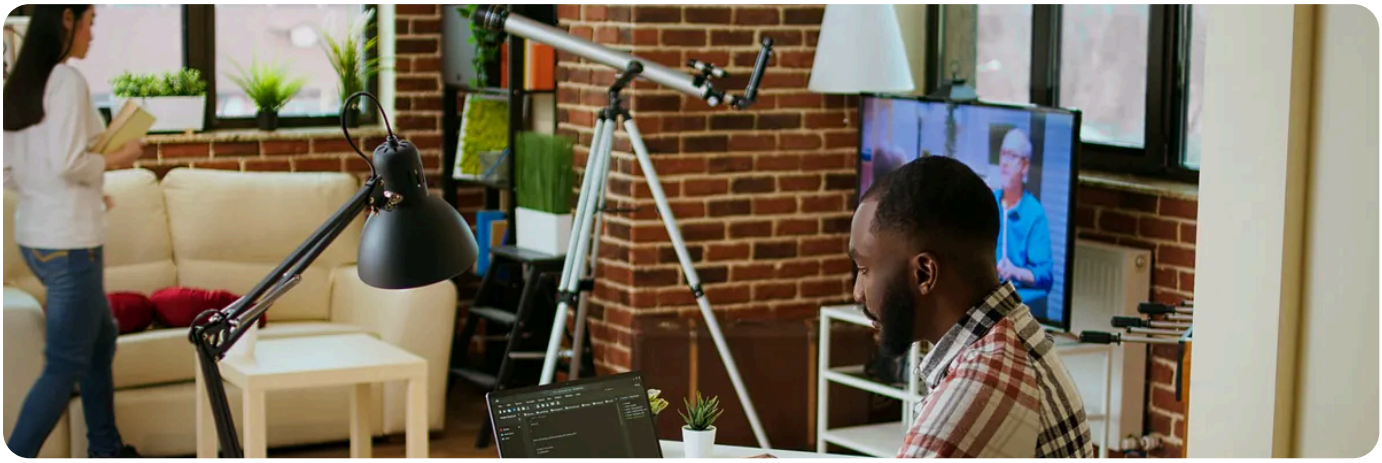
CODE	VERSION	COUNTRY · LANGUAGE
RLX-INT-KB-8670	v2026.11.26	International · English

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/8670

Using the OpenAI SDK with All in One AI: only the address and the key change

What You Can Automate Created on **November 26, 2026** 3 min read



In short: the All in One AI chat is compatible with the OpenAI API. A program that already uses the OpenAI SDK switches to our models by changing **only the address and the key**. That is why **AI reselling** works for people who already have code written.

Install the SDK and point it at us

1. Install the official SDK: `pip install openai` in Python or `npm install openai` in Node.js.
2. Create a key on the **API page in My MozOut**, with `ai:read` and `ai:write`. Use a `sandbox_` key while you write the program.
3. Keep the key in the `MOZOUT_API_KEY` environment variable.
4. Give the SDK the address `https://api.mozout.com/v1` and the key.

In Python:

```
import os
from openai import OpenAI

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

completion = client.chat.completions.create(
model="deepseek-ai/deepseek-v4-flash",
messages=[{"role": "user", "content": "Write a two-sentence welcome message for our
```

```
website."}],
max_tokens=200,
)
print(completion.choices[0].message.content)
```

In Node.js:

```
import OpenAI from "openai";

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

const completion = await client.chat.completions.create({
model: "deepseek-ai/deepseek-v4-flash",
messages: [{ role: "user", content: "Write a two-sentence welcome message for our website." }],
max_tokens: 200,
});
console.log(completion.choices[0].message.content);
```

The `model` always has a slash, as in the example. `GET /v1/models` gives the full list, with each model's price in credits, and filters by kind with `type`.

Streaming, so the text appears as it is written

With `stream=True`, the answer arrives in pieces as the model writes. Always ask for `stream_options={"include_usage": True}`: the token count comes in the last event, and without it the whole reservation is charged.

How each request is charged

Before calling the model, we reserve credits for the input and for the output `max_tokens`. Without `max_tokens`, the reservation counts 4096 output tokens. At the end, we charge the real tokens and return the rest to your balance. If the balance is not enough for the reservation, the answer is `402` and nothing is charged. With the Coding Plan, the plan's models spend points instead of credits.

> **Always set `max_tokens`.**
> It limits the length of the answer and lowers the reservation, which avoids a `402` when your balance is low.

Agents, with the same code

An agent built in All in One AI is used in the same chat. In `model`, send `agent/` followed by the agent's id. Do not send the instructions or the knowledge: the agent adds them itself. To see how far an agent can go, read **what an AI agent is**.

What the SDK does not do

The OpenAI SDK does not have the image, video and voice operations. For those, use plain HTTP requests to `POST /v1/media/generations` and check the status at `GET /v1/media/generations/{id}`. The coding tools work too: Claude Code uses the Messages format and Codex CLI the Responses format, with the same key.

Errors, in the SDK

The OpenAI SDKs read the message from `error.message` and raise an exception with the HTTP status. In Python it is `openai.APIStatusError`, with `status_code`.

Status	When	What to do
`402 insufficient_credits`	There are not enough credits for the reservation.	Buy credits in All in One AI or lower `max_tokens`.
`403 insufficient_scope`	The key does not have `ai:read` or `ai:write`.	Create a key with the right permission.
`404 model_not_found`	The model does not exist or is not a text model.	Check the `id` in `GET /v1/models`.
`429 rate_limited`	You went over the per-minute request limit.	Wait the number of seconds in `Retry-After`.
`502 upstream_error`	The AI service did not answer as expected.	Try again in a little while.

The full list is in **our API's error codes**.

RECOMMENDED

Already have code that uses the OpenAI SDK? See the AI reselling page and create your key.

See AI reselling →

SEE ALSO

- **The same API key for your clients: AI and the rest of our API with one key** →
- **Testing All in One AI in the Sandbox at no cost: what each answer looks like** →
- **Our API's error codes: what each one means, and which ones are worth retrying** →

What an AI agent is, and what it still cannot do reliably →

AI reselling in the Reseller Program →

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

Online version of this document

RLX-INT-KB-8670 · v2026.11.26 · Generated on 26/11/2026 10:24