Mozout

Generated on 29/11/2026 04:04

KNOWLEDGE BASE · PUBLIC DOCUMENT

Sending tools (function calling) in a chat request

Using the AI API

CODE	VERSION	COUNTRY · LANGUAGE
RLX-INT-KB-9102	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/9102

Sending tools (function calling) in a chat request

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



In short: tools (function calling) let the model ask your program to run a function, for example to look up an order, and then answer with the result. The model does not run anything: it only says which function it wants and with which values. Your program runs it. In chat, `POST /v1/chat/completions`, tools go in the `tools` field, in the OpenAI format.

How it works, in three steps

1. Your program sends the messages and the list of tools, each with a name, a description and its parameters (as JSON Schema).

2. If the model wants to use one, the answer comes with `finish_reason` `tool_calls` and, in `message.tool_calls`, the call `id`, the function name and the arguments in `function.arguments` (a JSON string).

3. Your program runs the function and makes a second request with the conversation, the model's message and a message with the `tool` `role`, the `tool_call_id` and the result in `content`. The model then answers in text.

The first request

```
{
  "model": "deepseek-ai/deepseek-v4-flash",
  "messages": [{"role": "user", "content": "What is the status of order 1042?"}],
  "tools": [{
    "type": "function",
    "function": {
      "name": "get_order_status",
      "description": "Returns the status of an order by its number.",
      "parameters": {
```

```
        "type": "object",
            "properties": {"order_id": {"type": "string", "description": "The order
number"}},
        "required": ["order_id"]
      }
    }
  }],
  "max_tokens": 300
}
```

The answer and the second request

The model asks for the function (part of the answer; the `id` is always the one in your own answer):

```
"message": {
  "role": "assistant",
  "content": null,
  "tool_calls": [{"id": "call_1", "type": "function",
    "function": {"name": "get_order_status", "arguments": "{\"order_id\": \"1042\"}"}}]
},
"finish_reason": "tool_calls"
```

The second request has the same `model` and the same `tools`, and these messages:

```
"messages": [
  {"role": "user", "content": "What is the status of order 1042?"},
    {"role": "assistant", "content": null, "tool_calls": [{"id": "call_1", "type":
"function",
          "function": {"name": "get_order_status", "arguments": "{\"order_id\":
\"1042\"}"}}]},
  {"role": "tool", "tool_call_id": "call_1", "content": "{\"status\": \"shipped\"}"}
]
```

The other OpenAI fields, such as `tool_choice`, go through as they are.

Which model to use

Use a model that accepts tools. The models recommended for **coding tools** all accept them, among them `moonshotai/kimi-k2.7-code`, `zai-org/glm-5.2` and `deepseek-ai/deepseek-v4-flash`. Before you go live, test the whole cycle with the model you chose.

> **Check the arguments before you run the function.**
> The model does not always generate valid JSON, and it can ask for values that make no sense. Validate them, and never let the model decide on its own things such as payments or deleting data.

Cost, Sandbox and the other formats

Each request pays for its own tokens: an answer that uses one tool makes two requests to the model. With a `sandbox_` key, chat always answers "Resposta de teste (sandbox).", so no tool calls appear. To test the cycle, use the `prod_` key with a cheap model and a low `max_tokens`. In the Anthropic format, the calls come as `tool_use` blocks and you answer with `tool_result`; in Responses, as `function_call` and `function_call_output` items.

RECOMMENDED

Want to pick a model that accepts tools? See the catalog in All in One AI.

See All in One AI →

SEE ALSO

- Apps made for Claude: the Anthropic Messages format at /v1/messages
- The OpenAI Responses format: /v1/responses and responses kept for 30 days
- Choosing a text model: price, context window and what each one is for
- What a chat request costs through the API: the reservation, max_tokens and the final adjustment
- 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

Online version of this document

RLX-INT-KB-9102 · v2026.11.28 · Generated on 29/11/2026 04:04