Function calling

- How can I use function calling?
- What models support function calling?

import { YouTube, TabItem, Tabs } from "~/components";

Function calling enables people to take Large Language Models (LLMs) and use the model response to execute functions or interact with external APIs. The developer usually defines a set of functions and the required input schema for each function, which we call `tools`. The model then intelligently understands when it needs to do a tool call, and it returns a JSON output which the user needs to feed to another function or API.

In essence, function calling allows you to perform actions with LLMs by executing code or making additional API calls.

How can I use function calling?

Workers AI has embedded function calling which allows you to execute function code alongside your inference calls. We have a package called `@cloudflare/ai-utils` to help facilitate this, which we have open-sourced on Github.

For industry-standard function calling, take a look at the documentation on Traditional Function Calling.

To show you the value of embedded function calling, take a look at the example below that compares traditional function calling with embedded function calling. Embedded function calling allowed us to cut down the lines of code from 77 to 31.

```
# The ai-utils package enables embedded function calling
npm i @cloudflare/ai-utils
```

```
import {
  createToolsFromOpenAPISpec,
  runWithTools,
  autoTrimTools,
} from "@cloudflare/ai-utils";

export default {
  async fetch(request, env, ctx) {
    const response = await runWithTools(
      env.AI,
      "@hf/nousresearch/hermes-2-pro-mistral-7b",
      {
        messages: [{ role: "user", content: "Who is Cloudflare on github?" }],
        tools: [
          // You can pass the OpenAPI spec link or contents directly
          ...(await createToolsFromOpenAPISpec(
            "https://gist.githubusercontent.com/mchenco/fd8f20c8f06d50af40b94b0671273dc1/raw/f9d4b5c
d5944cc32d6b34cad0406d96fd3acaca6/partial_api.github.com.json",
            {
              overrides: [
                {
                  // for all requests on *.github.com, we'll need to add a User-Agent.
                  matcher: ({ url, method }) => {
                    return url.hostname === "api.github.com";
                  },
                  values: {
                    headers: {
                      "User-Agent":
                        "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Ge
Chrome/112.0.0.0 Safari/537.36",
                    },
                  },
                },
              ],
            },
          )),
        ],
      },
    ).then((response) => {
      return response;
```

```
  });

  return new Response(JSON.stringify(response));
 },
};
```

```
export default {
 async fetch(request, env, ctx) {
  const response = await env.AI.run(
   "@hf/nousresearch/hermes-2-pro-mistral-7b",
   {
    messages: [{ role: "user", content: "Who is Cloudflare on GitHub?" }],
    tools: [
     {
      name: "getGithubUser",
      description:
       "Provides publicly available information about someone with a GitHub account.",
      parameters: {
       type: "object",
       properties: {
        username: {
         type: "string",
         description: "The handle for the GitHub user account.",
        },
       },
       required: ["username"],
      },
     },
    ],
   },
  );

  const selected_tool = response.tool_calls[0];
  let res;

  if (selected_tool.name == "getGithubUser") {
   try {
    const username = selected_tool.arguments.username;
```

```
        const url = `https://api.github.com/users/${username}`;
        const res = await fetch(url, {
          headers: {
            // Github API requires a User-Agent header
            "User-Agent":
              "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/112.0.0.0 Safari/537.36",
          },
        }).then((res) => res.json());
      } catch (error) {
        return error;
      }
    }

    const finalResponse = await env.AI.run(
      "@hf/nousresearch/hermes-2-pro-mistral-7b",
      {
        messages: [
          {
            role: "user",
            content: "Who is Cloudflare on GitHub?",
          },
          {
            role: "assistant",
            content: JSON.stringify(selected_tool),
          },
          {
            role: "tool",
            content: JSON.stringify(res),
          },
        ],
        tools: [
          {
            name: "getGithubUser",
            description:
              "Provides publicly available information about someone with a GitHub account.",
            parameters: {
              type: "object",
              properties: {
                username: {
```

```
□□□□□□□□□□type: "string",
□□□□□□□□□□description: "The handle for the GitHub user account.",
□□□□□□□□□},
□□□□□□□□},
□□□□□□□□required: ["username"],
□□□□□□□},
□□□□□□},
□□□□],
□□□},
□□);
□□return new Response(JSON.stringify(finalResponse));
□},
};
```

What models support function calling?

There are open-source models which have been fine-tuned to do function calling. When browsing our model catalog, look for models with the function calling property beside it. For example, @hf/nousresearch/hermes-2-pro-mistral-7b is a fine-tuned variant of Mistral 7B that you can use for function calling.

□□□ #2
□ □□□ □□□ 2026-11-01 05:28:25 UTC
□ □□□ □□□ 2026-11-01 05:28:28 UTC