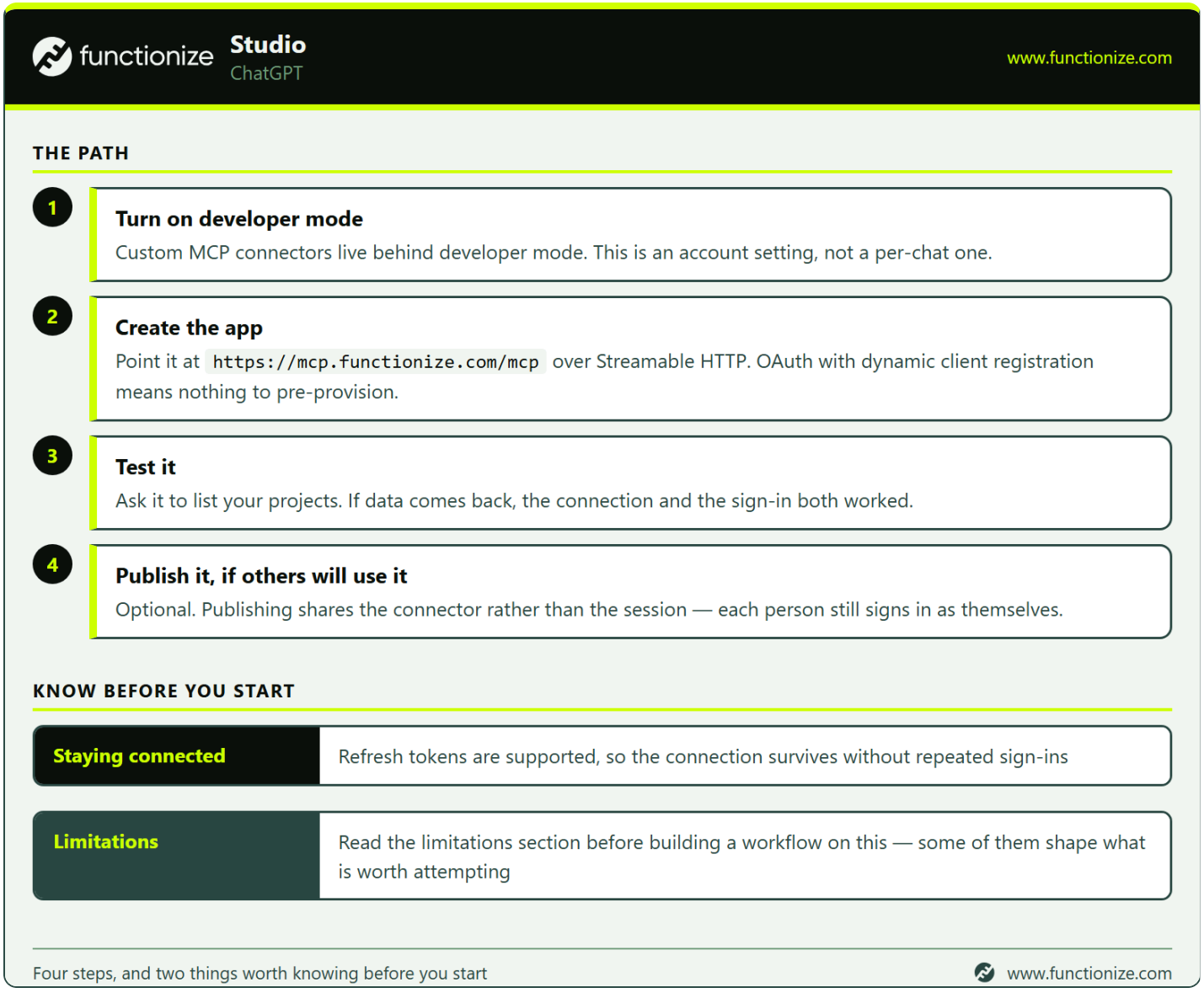
Connect ChatGPT to the Functionize MCP Server

Model Context Protocol (MCP) · https://docs.functionize.com/hc/en-us/articles/43755024399255-Connect-ChatGPT-to-the-Functionize-MCP-Server

ChatGPT can connect to the Functionize MCP server as a custom app. The setup is different from the other clients in this section: it is done entirely in the ChatGPT web interface, it requires a workspace plan, and it is controlled by your ChatGPT administrator rather than by you.



Supported, but not yet confirmed end to end. The configuration below is correct and the flow is standard, but we have not yet verified a complete round trip against our server on a ChatGPT workspace. If you get stuck, tell us what you saw rather than assuming you have it wrong.

What you need first

- **A ChatGPT Business, Enterprise or Edu workspace.** Custom MCP apps and developer mode are workspace features on ChatGPT web.
- **Pro is not enough.** On Pro, developer mode allows read and fetch actions only. Our session tools start sessions and send messages, which are write actions, so a Pro account cannot drive Functionize through this route.
- **Administrator involvement.** Only admins and owners can enable developer mode and publish an app. On Enterprise and Edu, an admin can grant developer access to specific people through role-based access control.
- **A Functionize account**, which you will sign in with during setup.

There is nothing to install. ChatGPT talks to remote HTTPS servers only, so no bridge and no Node.js are involved.

Step 1 — Turn on developer mode

Where this lives depends on your plan.

Business — an admin or owner enables it in **Workspace Settings** → **Permissions & Roles** → **Connected Data**, using the developer mode / create custom MCP connectors toggle. Each admin enables it for themselves; it is not applied to everyone at once.

Enterprise and Edu — admins and owners enable it from **Settings** → **Apps** → **Advanced Settings**. Admins can use role-based access control to grant developer mode to specific members, who then toggle it on for themselves in the same place.

Step 2 — Create the app

Admins go to **Workspace Settings** → **Apps** → **Create**. Authorized users go to their own **Settings** → **Apps** → **Create**.

Fill in the details:

Name	Functionize
MCP server URL	`https://mcp.functionize.com/mcp`
Authentication	OAuth

Then click **Scan Tools**. ChatGPT connects to the server and reads the available tools. You will be prompted to complete the Functionize sign-in during this step.

Sign in with the email and password you use for Functionize Studio. If you registered with Google, set a password via **Forgot password** on the Studio login page first. If your company signs in to Functionize through SSO, contact your Functionize representative.

Wait for the scan to finish, then click **Create**.

The app now appears as a draft under **Workspace Settings** → **Apps** → **Drafts**, and in your own **Settings** → **Apps** → **Enabled Apps** with a *Dev* label.

Step 3 — Test it

Open a new chat, select the draft app from the tools menu or mention it in your prompt, and ask it to list your Functionize agent sessions. ChatGPT asks you to confirm actions depending on the app's permissions, so expect a prompt before anything is written.

Step 4 — Publish it, if you want others to use it

Only admins and owners can publish. Go to **Workspace Settings** → **Apps** → **Drafts** and click **Publish**. Review the safety warnings, particularly around write actions. Once published, the app appears in the workspace's approved connectors and in members' Apps settings with a *custom* label.

On Enterprise and Edu, admins can use **Configure Actions** to choose exactly which tools the app may use, and **Configure Access** to limit it to particular groups. Both can be revisited afterward from the app's **Action control** menu.

A note on staying connected

OpenAI warns that if an MCP server's OAuth provider does not issue refresh tokens, ChatGPT can lose access once the original authorization expires, forcing people to sign in repeatedly.

This is not a problem with the Functionize server. We advertise `offline_access` in our OAuth metadata and support the `refresh_token` grant type, which is exactly what that requirement asks for. You can verify it yourself:

```
https://mcp.functionize.com/.well-known/oauth-authorization-server
```

Look for `offline_access` under `scopes_supported` and `refresh_token` under `grant_types_supported`.

Limitations worth knowing before you start

- **ChatGPT web only.** Custom apps are configured and used on the web client.
- **No local servers.** ChatGPT connects to remote HTTPS endpoints only, which suits our hosted server.
- **Business plans cannot edit a published app.** To change tools or metadata after publishing, recreate and republish it. Enterprise and Edu have finer controls, including refreshing the action list in place.
- **Updates are not automatic.** If we add tools to the server, an admin on **Enterprise or Edu** uses **Refresh** to pick them up, and new actions arrive disabled by default. On **Business**, a published app cannot be refreshed — recreate and republish it.
- **Team pinning is not available here.** OpenAI does not document custom request headers for custom apps, so the connection acts as **your default Functionize team**. If you need a specific team, use a client that supports the `X-Functionize-Team-Id` header — see Connect Any MCP Client to Functionize.

If something goes wrong

If the tool scan fails, check that you entered the URL exactly, including the `/mcp` path, and that you completed the sign-in prompt rather than closing it.

If the app connects but does not appear in a chat, confirm it is enabled in **Settings** → **Apps** and that you have selected it for that message — app selection applies per message, not for a whole conversation.

For anything else, see Troubleshooting the Functionize MCP Connection, or raise a ticket with your account email and the point at which it stops.

Related

- Getting Started with the Functionize MCP Server — what the server can do, once connected
- Connect Any MCP Client to Functionize — the endpoint, scopes and transport in one table
- Troubleshooting the Functionize MCP Connection — when it does not connect
- Exercise 7 - Drive Studio From Your Own AI Tool — ten minutes, against the demo store
- The Independent Tester Principle: Never Build and Test in the Same Session — why this belongs in a session of its own