Troubleshooting the Functionize MCP Connection

Model Context Protocol (MCP) · https://docs.functionize.com/hc/en-us/articles/43753415472535-Troubleshooting-the-Functionize-MCP-Connection

Start by working out which route you are on, because the causes are different.

functionize **Studio**
Troubleshooting MCP
www.functionize.com

START WITH WHAT YOU ARE SEEING

Connector is there, not working	Claude Desktop — usually a stale definition from an earlier attempt
Server not listed after restart	Bridge route. Check the absolute path to `npx` : Claude Desktop does not inherit your shell PATH
Nothing happens on restart	Bridge route. A window close is not a restart
No browser tab for sign-in	Bridge route. The OAuth callback needs a loopback address it is allowed to use
Sign-in page will not let you past	Account or team access, not the connection itself
Tools load, calls return nothing	A token problem. Sign in again or clear the stale one
A team you did not expect	The pinning header, or the absence of one — switching teams in Studio does not move the connection
Connected to an older build	An internal endpoint from an earlier setup is still in your configuration

Let Claude diagnose it. If you are already in Claude Code or Claude Desktop, paste the diagnostic prompt from this article — it inspects the configuration, tries the endpoint and tells you which of the above you have.

Start from what you are seeing, not from what you think is wrong

Find what you are seeing, then jump to that section.

- **Claude Desktop through Settings → Connectors** — the verified path. Almost everything here is fixed by removing the connector and adding it again.
- **Claude Code** — a native HTTP entry. Problems are usually scope or sign-in.
- **The `mcp-remote` bridge** — the fallback. This is where absolute paths, JSON syntax and cached tokens come in.
- **claude.ai in a browser** — there is no bridge there at all, so any advice below about `npx` and configuration files does not apply.

The two causes that apply to everyone are sign-in and team pinning. The rest belong to the bridge.

Claude Desktop: the connector is there but not working

Remove the connector and add it again. Settings → Connectors, delete the Functionize entry, then **Add custom connector** with `https://mcp.functionize.com/mcp` and sign in when the browser tab opens. That clears most problems on this route.

If **Add custom connector** is missing or disabled, you are on a Team or Enterprise plan where only Owners can add connectors. Ask an Owner in your organization.

Let Claude diagnose it for you

Before working through this by hand, try pasting this prompt into Claude Code. It checks the common causes in order and tells you what it found.

```
 My Functionize MCP connection is not working. Diagnose it. Work out my operating
system yourself and use the right paths.

1. Check the server is reachable. Expect HTTP 200 from:
   https://mcp.functionize.com/.well-known/oauth-protected-resource

2. Check node --version is 18 or newer.

3. Show me my current MCP configuration:
     Claude Code:    run "claude mcp list"
     Claude Desktop: read the config file
       macOS:   ~/Library/Application Support/Claude/claude_desktop_config.json
       Windows: %APPDATA%\Claude\claude_desktop_config.json

4. For any Functionize entry you find, check all of the following and tell me which
   fail:
   - Does the URL point at mcp.functionize.com, and NOT internal-dev.functionize.com?
   - Is "command" an ABSOLUTE path that exists and is executable, rather than a
     bare "npx"?
   - Is the whole config file valid JSON?

5. Tell me exactly what is wrong and what to change. Do not change anything yet -
   show me first.
```

Bridge fallback: the server is not listed after restarting

Almost always the `command` path in your Claude Desktop configuration is not actually executable.

A bare `"npx"` will not work. Claude Desktop launches commands without your shell, so anything provided by nvm, fnm or Volta is invisible to it. You need the full path — and on Windows, the `.cmd` file.

Find the real path:

macOS

```
which npx
```

Windows (PowerShell)

```
(Get-Command npx).Source
```

Then confirm the path you put in the config actually runs:

```
"/opt/homebrew/bin/npx" --version
```

The other common cause is invalid JSON — usually a missing or extra comma after adding the new entry. Paste the file into any JSON validator before restarting.

Bridge fallback: nothing happens when you restart Claude Desktop

The app was probably not fully quit. Closing the window leaves it running, and it will not re-read its configuration.

- **macOS:** press **Cmd+Q**.
- **Windows:** right-click the Claude icon in the system tray near the clock and choose **Quit**. If it is not in the tray, end the process in Task Manager.

Then reopen the app.

Bridge fallback: no browser tab opens for sign-in

Run the bridge by hand so you can watch the handshake and see any error:

```
/opt/homebrew/bin/npx -y mcp-remote@latest https://mcp.functionize.com/mcp
```

Use your own npx path. On Windows, run the equivalent with your `npx.cmd` path.

You reach the sign-in page but cannot get past it

This is the most common problem in larger organizations, and it is rarely a broken connection.

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.

The MCP server has no separate account and no separate password. It authenticates against your existing Functionize login, and that is also what determines which projects and teams the connection can reach.

Do not guess at a password — repeated failures can temporarily block sign-in. If you are not sure which route applies to you, ask your Functionize representative rather than trying variations.

Check the domain. The sign-in page is hosted at `functionize.us.auth0.com`, and the server behind it is `mcp.functionize.com`. Both are ours. If you are ever asked for Functionize credentials on any other domain, do not enter them — send it to us instead.

Nobody at Functionize will ever ask you for that password. You only ever type it into that sign-in screen.

You want to sign in again, or clear a stale token

On the connector route, remove the connector and add it again; that re-runs sign-in.

On the bridge, tokens are cached on disk separately from your configuration file. **Quit Claude Desktop before deleting the cache**, or the running app keeps holding the stale credentials.

macOS

```
rm -rf ~/.mcp-auth
```

Windows (PowerShell)

```
Remove-Item -Recurse -Force "$env:USERPROFILE\.mcp-auth"
```

Then restart Claude Desktop. This clears cached tokens for every server bridged through `mcp-remote`, so you will be asked to sign in to those again too.

In Claude Code, run `/mcp`, select `functionize-hosted`, and re-authenticate.

You connected to an older internal build

An entry pointing at `internal-dev.functionize.com` will not work from outside our network. Remove it and add the production server at `mcp.functionize.com`.

In Claude Code:

```
claude mcp list
claude mcp remove <name>
```

In Claude Desktop, delete that entry from the configuration file. The old `@functionize/mcp` stdio package has been retired, so remove any entry for that too.

Clear the cached tokens as above, then quit and reopen the app.

Tools load but every call returns nothing, or says unauthorized

Two causes, in this order:

1. **The connection is pinned to a team you are not in.** Check for an `X-Functionize-Team-Id` header in your configuration. Remove it and reconnect: if calls start working, the team ID was the problem.
2. **The account may not be provisioned for agent sessions.** If there is no team header and calls still return nothing, raise it with us, including your account email and roughly when you signed in.

You are seeing a team you did not expect

By default the connection acts as **your default team**. Switching teams inside Studio does not move the MCP connection.

To pin a specific team, add the `X-Functionize-Team-Id` header as shown in the connection guides. Interactive clients read their configuration when the connection starts, so restart the connection after changing it.

Do not use `list_agent_teams` to confirm a pin. It echoes a pinned team ID back without checking it, so the ID appearing in its output tells you nothing about whether the pin took effect. Confirm it by listing sessions and checking you see the sessions you expect for that team.

On Windows through the bridge, write the header with no space after the colon — `X-Functionize-Team-Id:982`. A space is swallowed by the way arguments reach `npx`, and the header arrives malformed.

claude.ai in a browser

There is no local bridge in a browser, so nothing on this page about `npx`, absolute paths or configuration files applies there.

Use **Claude Desktop** and add the server under **Settings** → **Connectors** → **Add custom connector**. That is the path we have verified end to end.

Still stuck

Raise a ticket and include: which client and operating system you are using, the exact entry from your configuration with any team ID redacted, what `node --version` reports, and the point in the process where it stops. Never include your password.

Related

- Getting Started with the Functionize MCP Server — start here if nothing is set up yet
- Connect Any MCP Client to Functionize — the endpoint and OAuth details to check against
- Connect Claude Desktop to the Functionize MCP Server — the client most of these failures come from