Connecting the Encoding Music MCP Server

MCP server URL: `https://encoding-music.drdanielrb.com/mcp`

This guide connects one public MCP server to two AI clients: **Claude.ai** (commercial, hosted) and **LM Studio** (free, runs open models like Gemma locally). Pick whichever you have. If a few people get this running, others can follow along.

What this MCP server does

The **Model Context Protocol (MCP)** is an open standard that lets an AI assistant call external tools. The Encoding Music server is a toolbox for computational analysis of encoded scores (MEI, MusicXML, MIDI), built on music21 and CRIM Intervals. Once connected, you ask questions in plain language and the assistant calls the right tool. Available tools include:

Key & notes	analyse key, list notes, weighted note distribution
Intervals	melodic and harmonic intervals between voices
Patterns	melodic n-grams, n-gram matches, first occurrences, heatmaps
Counterpoint & form	cadence detection, sonority / voice-leading progress
Notation & audio	render notation, highlight notes, play excerpts
Files	list and register the available MEI scores on the server

Note: This is a public, internet-hosted server, so it works with cloud clients (Claude.ai connects from Anthropic's servers, not your laptop) and local clients (LM Studio). No password or API key is needed to connect.

Option A: Claude.ai (commercial)

You need: a Claude account (Free, Pro, or Max). Custom connectors are in beta; Free accounts may add one custom connector, Pro/Max have no limit.

1. Sign in at **claude.ai**.
2. In the left sidebar, open **Customize > Connectors** (or go straight to `claude.ai/customize/connectors`). The old Settings > Connectors page now just points here.
3. Click the **"+"** at the top-right of Connectors, then **"Add custom connector."**

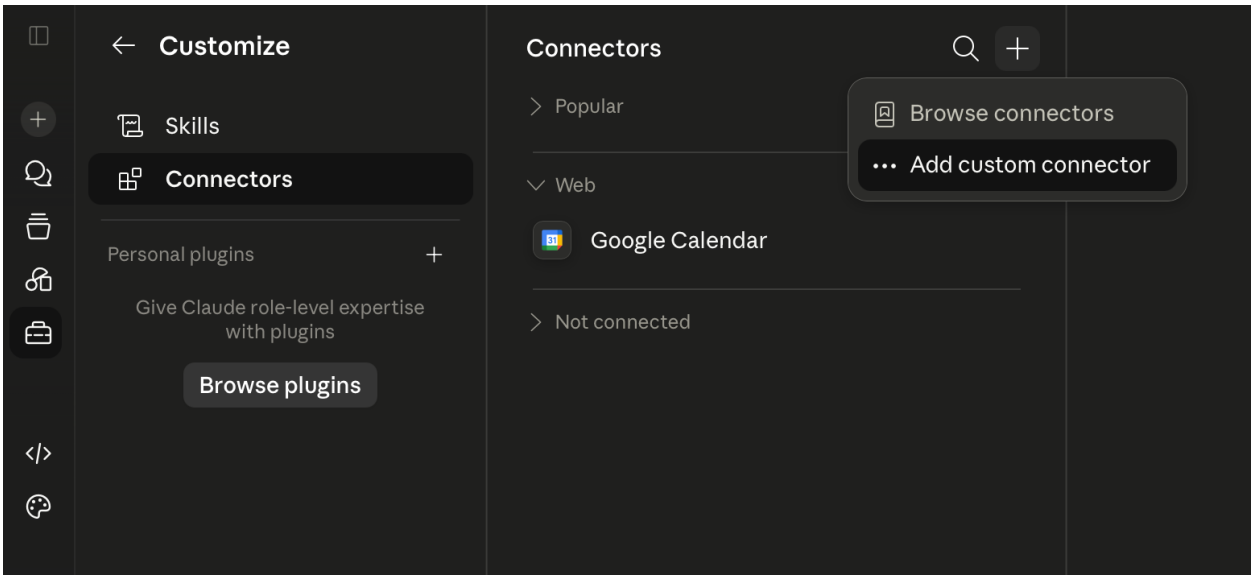

Figure 1. Customize > Connectors. The "+" (top-right) opens Add custom connector. (Reach Customize from the briefcase icon in the left sidebar.)

4. Give it a **name** (e.g. Encoding music) and paste the URL into **Remote MCP server URL**: `https://encoding-music.drdanielrb.com/mcp`
5. Leave **Advanced settings** (OAuth Client ID / Secret) blank, since this server doesn't need them. Click **Add**.

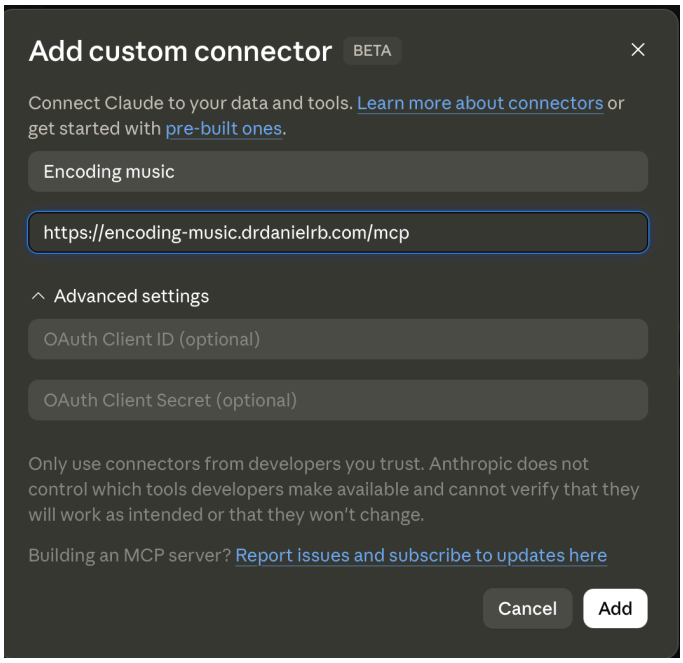

Figure 2. The Add custom connector dialog: enter a name and the server URL, then click Add.

6. After it connects, the connector appears with a **Custom** label. On its page you can set **Tool permissions** per tool: Allow, Needs approval, or Deny.

7. Open a chat. Click the **"+"** at the lower-left of the message box, choose **Connectors**, and make sure **Encoding music** is on for that conversation.
8. Test it: "List the available MEI files," then "Analyse the key of [one of them]." Approve the tool call when Claude asks.

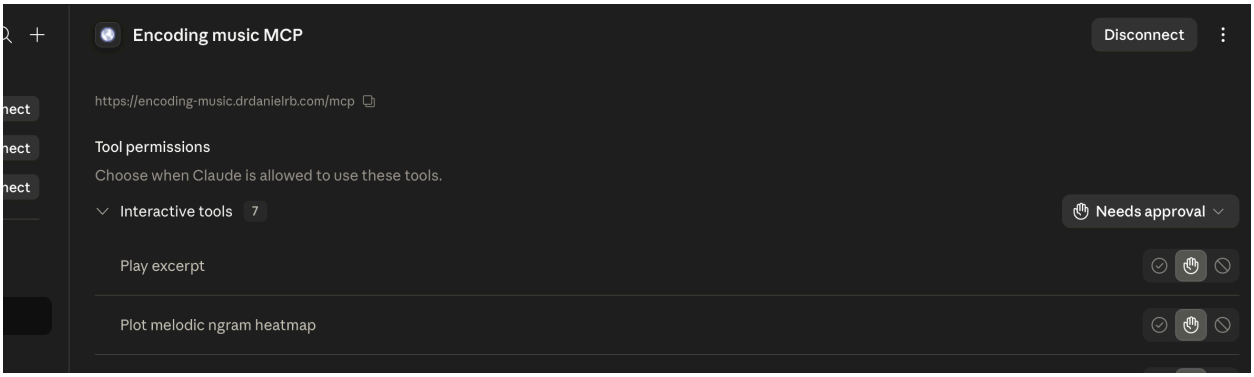

Figure 3. The connected Encoding Music connector with its tool permissions.

Claude Desktop: you can connect the same server there too. The configuration uses the same kind of JSON shown for LM Studio below. Guide: modelcontextprotocol.io/docs/develop/connect-local-servers

Option B: LM Studio with Gemma (free / local)

You need: LM Studio **0.3.17 or newer** (these steps were verified on 0.4.17). Download from `lmstudio.ai` if needed. Newer builds noticeably improve Gemma's tool-call reliability.

A note on local models: these small models are perfect for getting a feel for how MCP tools work, but they may lose track and fill up the context window faster than a cloud model, so expect the occasional missed tool call or a need to start a fresh chat.

Step 1: Download & load a Gemma model

1. Press **Cmd+Shift+M** (Mac) or **Ctrl+Shift+M** (Windows/Linux) to open model search.
2. Search **Gemma** and pick an instruction-tuned build whose capabilities list **Tool Use**; that tag is what matters here. Two solid choices: Gemma 4 12B Instruct (most reliable for the live demo) or Gemma 4 E4B (lighter, for modest laptops). Pick the largest your RAM comfortably handles, then click **Download**.
3. When it finishes, **load** the model from the selector at the top (or bottom) of the chat.

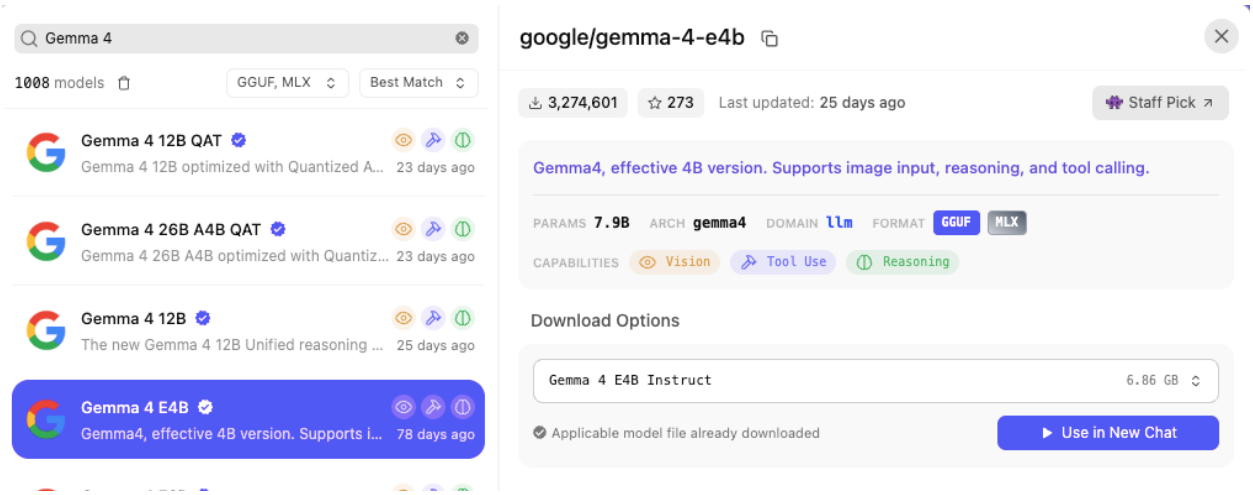

Figure 4. Model search (Cmd/Ctrl+Shift+M). Choose a Gemma 4 instruct build whose capabilities include Tool Use, then Download.

4. When loading, set **Context Length to ~16K (16384) or higher**. MCP tool definitions and results fill the context quickly, and the small default context makes tool calls quietly stop. 16K is a good balance; larger uses more memory and, if it spills off the GPU, slows everything down.

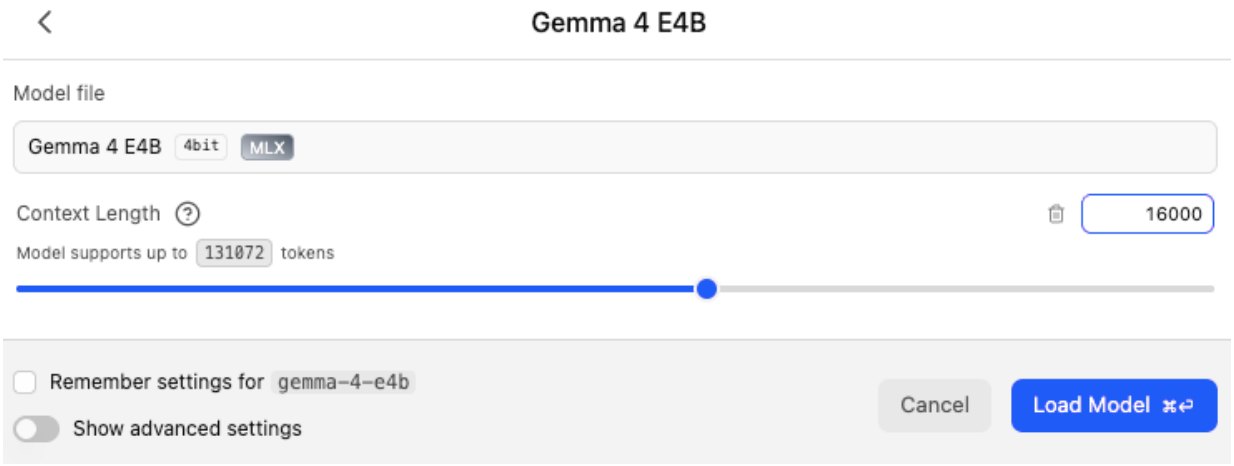

Figure 5. Set Context Length in the model's load settings (here 16K). Gemma 4 supports far more, but bigger reserves more memory, so raise it only as needed.

Step 2: Add the MCP server (edit mcp.json)

1. In a chat, click the **tools** (wrench) icon in the message composer to open the **Integrations** popover.
2. Click the **"+"** at the top of that popover, then **MCP > Edit mcp.json**.

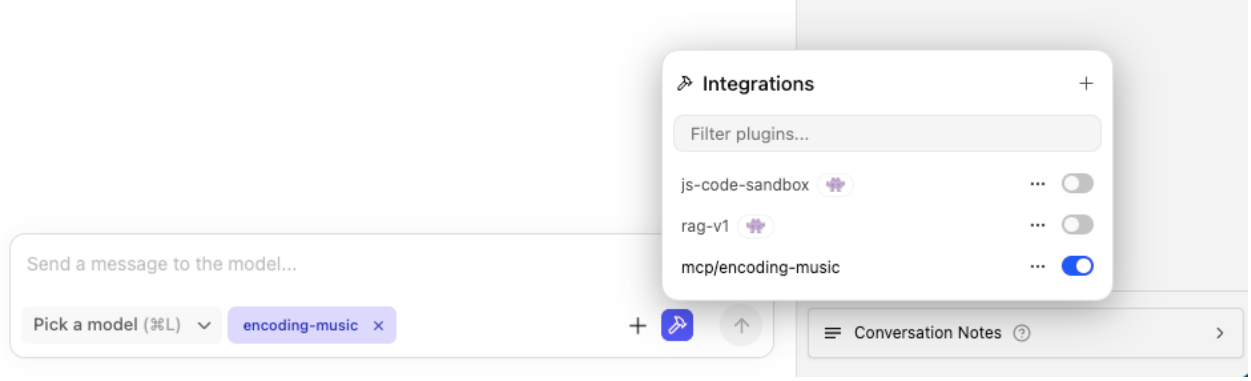

Figure 6. LM Studio: the composer's tools icon opens Integrations. Use "+" > MCP > Edit mcp.json. Each installed server also has an on/off toggle here.

3. Add the Encoding Music server inside `mcpServers`, then save (Cmd+S / Ctrl+S). LM Studio reloads and the tools appear:

```
{
  "mcpServers": {
    "encoding-music": {
      "url": "https://encoding-music.drdanielrb.com/mcp"
    }
  }
}
```

If you already have servers listed, just add the `"encoding-music": { ... }` entry inside the existing `mcpServers` block, separated by a comma.

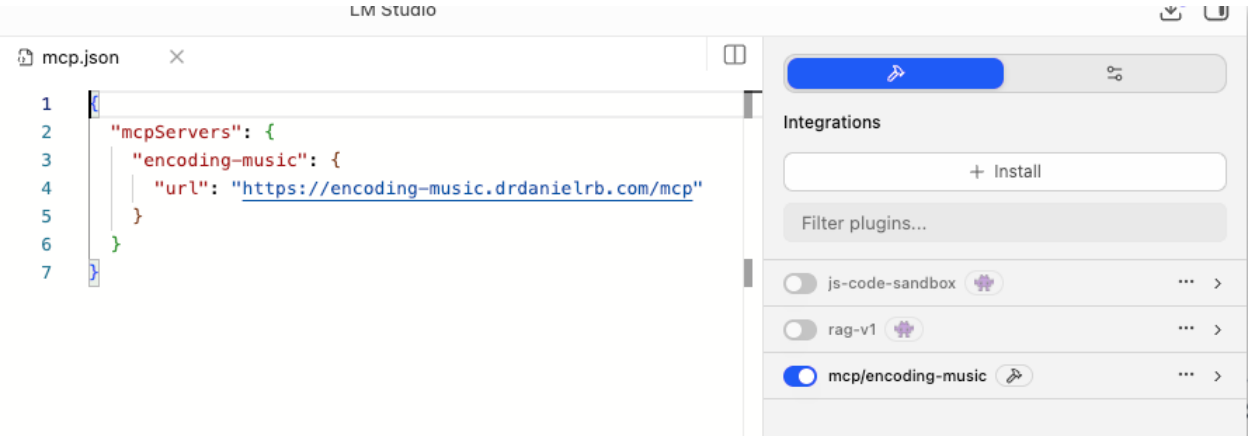

Figure 7. The mcp.json editor with the Encoding Music entry. LM Studio shows a safety caution because MCP servers can run code, so only add servers you trust.

Step 3: Use it with Gemma

1. In the **Integrations** popover, make sure **mcp/encoding-music** is toggled on.
2. Ask something that needs a tool, e.g. "Using the encoding-music tools, list the available MEI files."
3. When Gemma wants to run a tool, LM Studio shows a confirmation, so click **Allow**. (You can pre-approve specific tools under Settings > Integrations.)

4. Follow up: “Analyse the key of that piece and show the notation.”

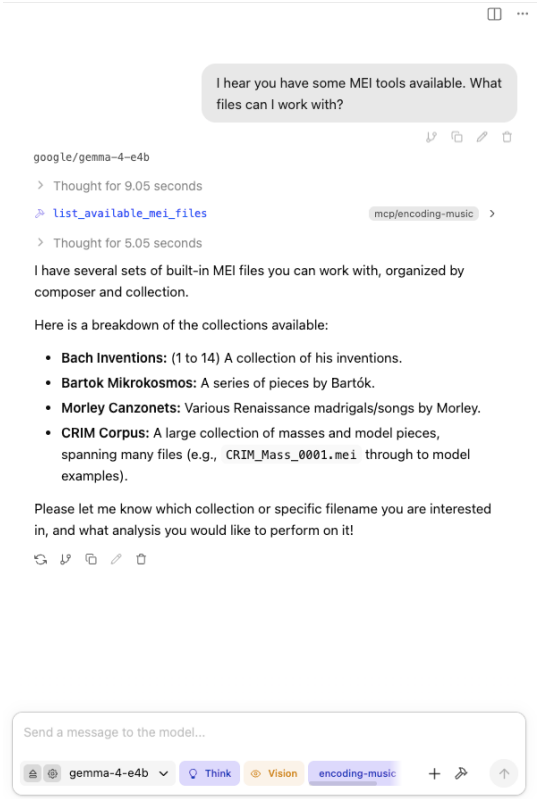

Figure 8. Gemma calling a tool from the Encoding Music server (here list_available_mei_files) and summarising the result.

Tips for local models: smaller models call tools less reliably than Claude. If a tool isn't triggered, name it explicitly (“call list_available_mei_files”). Some tools return large outputs that can overflow a local model's context, so ask for one piece or one analysis at a time. Local models are improving rapidly so these limitations should lessen over time.

Troubleshooting

Claude.ai can't reach the server	Check the URL is exactly the one above. Cloud clients connect from Anthropic's servers, so the server must be on the public internet (this one is).
No tools show up in LM Studio	Check mcp.json is valid JSON (no trailing commas), re-save, and confirm you're on 0.3.17+.
Gemma ignores the tools	Make sure the model is loaded and is an instruct build; try naming the tool in your prompt, and confirm the toggle is on.
Tool calls stop after the first result	The context window filled up. Raise Context Length to 16K+ when loading the model and start a fresh chat; tool definitions plus a result overflow small windows.
A tool call seems stuck	Approve the confirmation prompt. Tools won't run until you allow them.