

MagertronTM

Claude Code Integration Guide

Point Anthropic's terminal agent at a governed gateway
Developer Edition

Connecting Claude Code to Magertron

Claude Code is Anthropic's terminal agent. Point it at Magertron and every tool call it makes is authorized per call, metered against your contracts, and recorded — the same as any other agent on the platform.

This takes about five minutes.

Before you start

Node.js 18 or later. Claude Code installs from npm.

A Magertron account that can reach the Developer Portal, with at least one server whose tools you are entitled to call. The portal only lists servers you can actually use; if it shows nothing, you have no grants yet and your administrator needs to give you some.

An Anthropic account or API key. Claude Code pays Anthropic for its own model usage. That is separate from Magertron, which governs and meters the *tools* — not the model you chose to drive them with. You will see Claude Code's conversation costs on your Anthropic bill and the tool calls in Magertron's billing dashboard.

1. Install Claude Code

```
npm install -g @anthropic-ai/claude-code
claude --version
```

Anything from 2.1.202 onward reports configuration problems clearly. Older versions report a missing `type` field as `command: expected string, received undefined`, which is confusing enough to be worth upgrading to avoid.

2. Get your configuration from the portal

Open the Developer Portal and click **Add to IDE**.

In the dialog, choose **Claude Code** from the dropdown at the top right. This matters: Claude Code's format differs from Cursor's in two ways that both fail silently if you use the wrong one.

- Each entry needs `"type": "http"`. Without it Claude Code reads the entry as a local command, skips it, and tells you the server has a url but no type.
- The token reference is `${MCP_TOKEN}`, not `${env:MCP_TOKEN}`. Claude Code does not expand the second form, so it would send the literal text as your bearer token and every call would fail with a 401 that looks like a permissions problem.

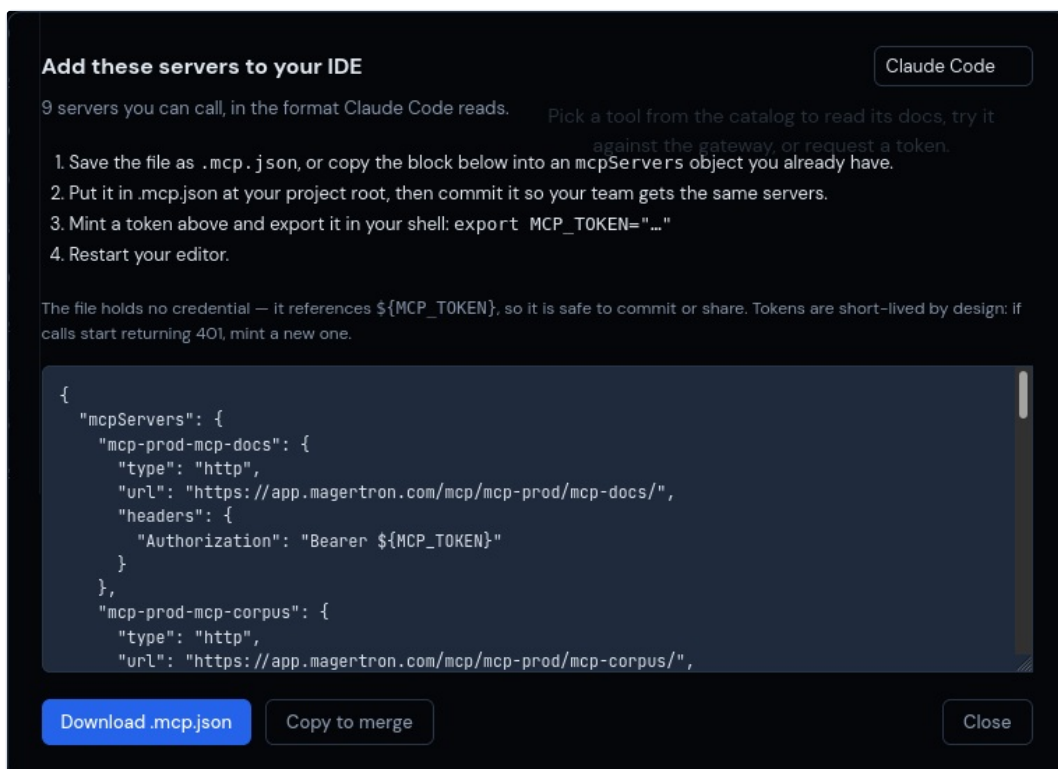


Figure 1 — The **Add to IDE** dialog with Claude Code selected. Note the `"type": "http"` entries and the bare `${MCP_TOKEN}` reference — this is the Claude Code format, not Cursor's.

Click **Download .mcp.json**.

⚠ **Your browser will probably save it as `mcp.json`, without the leading dot.** Most browsers refuse to write a dotfile. Rename it when you move it — the next step does.

3. Put it where Claude Code looks

Claude Code reads `.mcp.json` from the root of the directory you start it in.

```
cd ~/your-project
cp ~/Downloads/mcp.json .mcp.json
```

Commit it if you want your team to get the same servers. It holds no credential — only a reference to an environment variable — so it is safe in version control.

4. Mint a token and export it

In the portal, mint a token. Then, in the shell you will run Claude Code from:

```
export MCP_TOKEN="paste-your-token-here"
```

Put it in your shell profile if you want it to persist. It must be a real environment variable — remote MCP servers cannot read a `.env` file.

The token is yours. Every call Claude Code makes carries it, so the audit trail shows those calls as *you*, not as some shared service identity. If someone asks who ran a query, the answer is a person.

⚠ **Tokens are short-lived by design.** When calls start failing with 401, mint a new one. That is the system working, not something breaking.

5. Start Claude Code and approve the servers

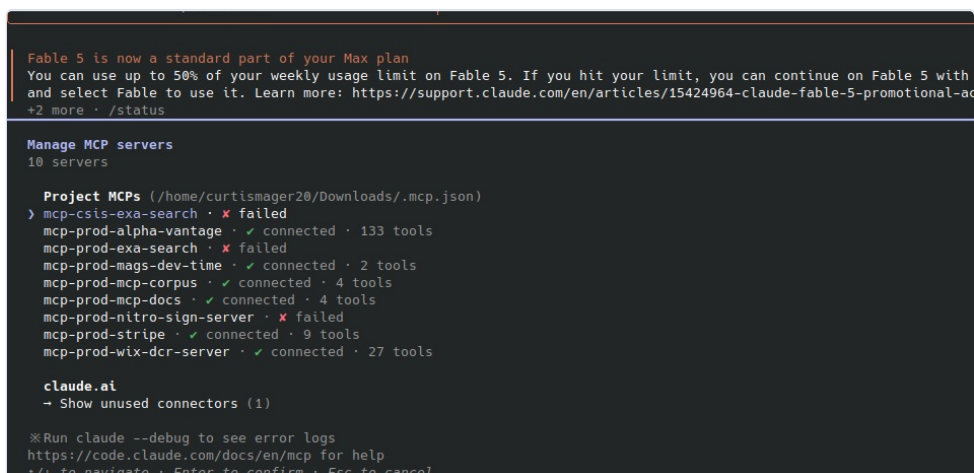
```
cd ~/your-project
claude
```

On first run it asks you to pick a theme and sign in to Anthropic. Then, because the servers come from a project file rather than your own configuration, it asks you to approve them. This is Claude Code's own safety step — a repository you cloned cannot silently point your agent at a server you have never seen.

Approve them, then check:

```
/mcp
```

Each server shows as connected, needing authentication, or failed.



```
Fable 5 is now a standard part of your Max plan
You can use up to 50% of your weekly usage limit on Fable 5. If you hit your limit, you can continue on Fable 5 with
and select Fable to use it. Learn more: https://support.claude.com/en/articles/15424964-claude-fable-5-promotional-ad
+2 more · /status

Manage MCP servers
10 servers

Project MCPs (/home/curtismager20/Downloads/.mcp.json)
> mcp-csis-exa-search · ✗ failed
mcp-prod-alpha-vantage · ✓ connected · 133 tools
mcp-prod-exa-search · ✗ failed
mcp-prod-mags-dev-time · ✓ connected · 2 tools
mcp-prod-mcp-corpus · ✓ connected · 4 tools
mcp-prod-mcp-docs · ✓ connected · 4 tools
mcp-prod-nitro-sign-server · ✗ failed
mcp-prod-stripe · ✓ connected · 9 tools
mcp-prod-wix-dcr-server · ✓ connected · 27 tools

claude.ai
→ Show unused connectors (1)

※Run claude --debug to see error logs
https://code.claude.com/docs/en/mcp for help
↑↓ to navigate · Enter to confirm · Esc to cancel
```

Figure 2 — `/mcp` listing the servers from the project file, each with its own state. A server that failed is that server's problem; when every one of them fails at once, look at your token instead.

6. Use it

Ask for something that needs a tool:

```
What time is it in Tokyo?
```

Claude picks the tool, calls it through Magertron's gateway, and answers. What happened underneath:

- 1. Claude Code sent a `tools/call` to `https://your-magertron/mcp/{namespace}/{server}/`
- 2. The gateway authorized it against your token's roles — per call, not per session
- 3. If the server's upstream needs a vendor credential, the gateway injected it. You never held it
- 4. The call was metered against that server's contract
- 5. It was written to the audit trail with your name on it

When something does not work

A server shows "Pending approval"

Run `claude` interactively in that directory and approve it. `claude mcp list` cannot approve on your behalf.

A server shows "Failed to connect"

Run `claude mcp get <server-name>`. The `Issue:` line gives the HTTP status and whatever the server said.

What you see	What it means
401	Your <code>MCP_TOKEN</code> is missing, expired, or not exported in this shell. Check with <code>echo \${MCP_TOKEN:0:12}</code>
403	Magertron refused. You are connected and authenticated but not entitled to that server or tool — ask your administrator for a grant
<code>upstream connect error</code>	The server's own backend is unreachable. Nothing to do with your setup; tell whoever owns that server

Every server fails at once

Usually the token. Check it is exported in the shell you started `claude` from, not a different one.

Claude says a tool is not available

Two possibilities, and they look the same from the outside. Either you have no grant for it, or your administrator has not yet approved that tool's definition. Magertron refuses calls to tool definitions that have changed since a human last reviewed them, which is what stops a server quietly widening what it can do after you connected to it.

Both are resolved by your administrator, and the audit trail shows which one it was.

What Magertron adds

You could point Claude Code straight at these servers and skip Magertron entirely. What you would lose:

Vendor credentials stay out of your hands. The gateway holds them and injects them per call. You configure a Magertron token, not an Exa key and a Stripe key and a GitHub token.

Every call is authorized when it happens, against your roles as they are at that moment. A grant revoked five minutes ago is already gone — you do not reconnect for it to take effect.

Cost is attributed to you. Not to a shared key that nobody can account for.

There is a record. For any call: who made it, what the rules were at the time, and what the gate decided. That record is signed and reconstructed from the audit trail on read, so it is not a log entry somebody could have edited afterwards.

Terminal notes

Claude Code redraws its interface as it works, which some terminals handle better than others.

In **Konsole**, copy and paste are `Ctrl+Shift+C` and `Ctrl+Shift+V` — plain `Ctrl+C` sends an interrupt. Selecting with the mouse usually copies automatically.

For a single question without the interactive interface:

```
claude -p "what time is it in Tokyo?"
```

That prints an answer and exits. Note it cannot approve project servers, so run `claude` interactively once first.

There are also Claude Code extensions for VS Code and JetBrains, which run the same agent inside your editor. The configuration is identical — the same `.mcp.json` in the same place.