PLUG IN TOOLS

Youth AI Lab · Workshop 4 of 6

Build your own chatbot



Co-funded by the European Union

❓ The question of the series

A chatbot only answers from **what it reads** before it answers. Every workshop of the series is **one answer** to the same question.

> **What enters the model's context, when, and who decides?**

📥 **What?**
rules, documents, search results, live data

⏱️ **When?**
always, at startup, or for each question

🤦 **Who decides?**
you, a search engine, or the model itself

🧭 Today: what happens when the chatbot acts outside?

1. Instructions	2. Knowledge, RAG	3. Search, skills	**4. Tools** today	5. Wiki	6. Fine-tuning
🎭	📚	🔍	📡	🗺️	🧪

Documents: what was true the day they were written.

Mode C: the model searches them itself.

Mode D: + **tools**, for what changes every day.

🗓️ The plan: 2 hours

👋 Welcome	📡 Our tool	🧩 Existing servers	🔍 Read before you install	🏁 Closing
20 min	20 min	30 min	25 min	15 min

📄 Follow on your screen: the `README.md` of your Codespace, or the one-page sheet.

💻 Your workspace: one Codespace per pair

A **Codespace** is VS Code in your browser: **nothing to install**. Everything you need is already inside.

> ⏱️ **Before you open it**, once per GitHub account: **github.com/settings/codespaces → Default idle timeout → 120 → Save**. It only applies to the Codespaces you create after.

1. Open the **Codespace link** of the workshop, at the top of its README
2. In the terminal, type `workshop-login` and paste the key of your pair
3. Wait for **Key OK**, then type `workshop bare`. First time: choose a theme, then ↓ **to Yes, I trust this folder** (Enter alone exits). To leave: **Ctrl+C twice**
4. Then type `workshop-start` and the number of the workshop. With a save from last time, add `--from my-work.zip`

> ⚠️ **The key is not shown** while you paste it: that is normal. If the Codespace stops after a long pause, reopen it: **nothing is lost**.

💻 No Codespace? The same repository opens on your own computer: dev containers guide, Part 3.

🐳 Bring your work

1. Drag `my-work.zip` into the file list (the newest: maybe `my-work (1).zip`, then its exact name between quotes), then `workshop-start 4 --from my-work.zip`. Check the `saved on` date.
2. No save? `workshop-start 4`: Feurisson, and the document of Benevento added for you.
3. Have the **official page of the fire danger level of your territory** ready, if it has one.
4. Second terminal: `openwebui start`. Do not wait for it: you will use it at the closing.

> 🚫 **Never make port 3000 public.** This Open WebUI has no accounts: anyone with the link could use your key.

🌡️ The gauge: how full is the context?

The **context** is everything the model reads before it answers. Its maximum size is the **context window**. A line above the input box shows **how full** it is.

🪟

200,000 tokens

the size of the window (a token is a piece of a word)

💶

Every message pays

for everything in the window

🤝

Same model for all

so that the pairs can compare

👁️ **Watch the gauge** at every change: it shows what entered the context.

📏 How to score an answer

Each answer gets **0, 1 or 2**. The score rewards **honesty** as much as knowledge.

2

Right, and it says **where it comes from**. Or "I don't know" when nobody knows.

1

Honest but not enough: no source, only partly right, or "I don't know" although the answer exists.

0

Wrong or **invented**.

💡 An **invented answer** is worse than **"I don't know"**.

📡 Our tool

Today's level: only a tool can know it

20 min · mode `D` · `/mcp`

📡 A tool brings back a fact

🧠

The model decides

to call the tool, from its description

📡

The tool answers

from `mcp/fire-danger.json`

🧑

A person

fills in the file, from the official source, on the day

MCP: one common way to plug a tool program (a **server**) into Claude Code, OpenCode or another chatbot.

🎬 Your turn: question 8, then Antwerp

1. `workshop D`, then `/mcp`: `fire-danger` and `time`. **Esc**.
2. Question 8, then **Ctrl+O**: what did the tool send back? Look for `why_no_level`.
3. `mcp/fire-danger.json`: today's `date`; for Antwerp, `status`, `source_says`, and `level`, `advice` only if `live`.
4. Question 8 again, **Ctrl+O**: what changed?

⚠️ Never make up a level: only what the official source says, on the day.

🧩 Servers that already exist

What does each one read, and send away?

30 min · `time` · `open-meteo` · `WORKSHOP_WEATHER=1`

🧩 Three servers side by side

Server	Who wrote it	What it reads	What the server sends on the internet
`fire-danger`	the workshop	`mcp/fire-danger.json`	nothing
`time`	the MCP project (official example)	the clock of the Codespace	nothing
`open-meteo`	a volunteer (its code is public)	the Open-Meteo service	the place and its coordinates

💡 What a server does is in its code and how it starts, not in its name.

🎬 Your turn: the time, then the weather

1. Open `.mcp.json` (`time`: `--local-timezone Europe/Paris`). Then `workshop D`: `Has the Pla Alfa map of Catalonia already had its morning update today?` → **Ctrl+O**: was `time` called?
2. The same, starting with `Use the time tool:`. Compare.
3. `WORKSHOP_WEATHER=1 workshop D`, `/mcp`: `open-meteo 17 tools` (the model gets 2).
4. `Will it be windy tomorrow in La Rochelle?` → **Ctrl+O**: what was sent? Then `Given the weather, what is the fire danger level today in La Rochelle?`

📏 Measured (2026-10-30): the weather plugged in but not called: **7,845** → **13,868 tokens** on fire danger questions, about 6,000 more.

🔍 Read before you install

and give your source

25 min · `mcp/fire_danger_server.py` · `mcp/fire-danger.json`

🔍 Before you install a server: five questions

1️⃣ **Who wrote it**, and is it still maintained?

2️⃣ **What does it read?** Its code, and its options in `.mcp.json`.

3️⃣ **What does it send away**, and to whom?

4️⃣ **Does it write or delete anything**, or send messages?

5️⃣ **Is its version fixed?**

🔑 A server runs **with your rights**. The description says what it is for; only the code says what it does.

📚 Tools in a chatbot of your own: your chatbot in Open WebUI (end of this workshop) · another way: OpenCode guide, Part 6 - Tools with MCP · Other MCP servers (its launcher accepts only small tools of its own folder)

🕵️ What a tool sends back enters the context

A web page, a document, an answer from a service can carry **instructions** that nobody wrote in your `AGENTS.md`.

Watch: a page that hides one sentence for AI assistants. What does the chatbot do with it?

❓ Who wrote this sentence, and who read it?

🎬 Your turn: your territory, with its source

1. `mcp/fire_danger_server.py` : the five questions. Does it call the internet?
2. `mcp/fire-danger.json` : the entry of your territory (or a copy of one entry, with its own name, `scale` , `source` , `season`).
3. `status` , `source_says` ; `level` and `advice` only if `live` . Save.
4. `workshop D` : `What is the fire danger level today in <your territory>?` Does it name your source?
5. `Add a field "checked_by"…` → **Edit file** → **3. No**: a change to a file waits for your yes.

✅ "There is no official level today, here is why" is a right answer.

🏁 Closing: column D, your chatbot, your save

1. **Predict**: which of 8, 11, 12 go up from C to D? Leave the conversation (**Ctrl+C twice**), then `replay D` and `score D 8 11 12`. Compare the rows with column C, not the Total (no C score for 8: `score C 8`).
2. Second terminal: `openwebui chatbot --name "<your chatbot>"`, then question 8 in Open WebUI. Click **View Result from fire-danger_fire_danger**: the same tool, read-only.
3. `workshop-save` → **Download...** `my-work.zip`: your `fire-danger.json` and everything before. If your computer already has one, bring the newest (maybe `my-work (1).zip`).

> 📝 **Exit card**, alone: (1) Why can only a tool give today's fire danger level? (2) What can a server do in your Codespace, and how do you check it? (3) Weather forecast or official level: what is the difference?

🆘 If you are stuck

Every command explains itself. From workshop 3, a few **slash commands** inside the chatbot show what it is using.

❓ In the terminal

`workshop help`

`replay --help`

`score --help`

🔍 In the chatbot

`/context` what fills the window

`/status` the key in use

`/mcp` the tools (from workshop 4)

Workshops 1 and 2: they answer "Unknown command". Watch the gauge above the input box.

🤦 Something breaks?

Ask the facilitator. Your files **stay in the Codespace**.

💡 To leave a conversation: **Ctrl+C twice**. In an open `.md` file, to read it as a page: **Ctrl+Shift+V** (**Cmd+Shift+V** on a Mac). In the terminal, copy / paste: **Ctrl+Shift+C / Ctrl+Shift+V** (**Cmd+C / Cmd+V** on a Mac). To change mode: `workshop <mode>` .