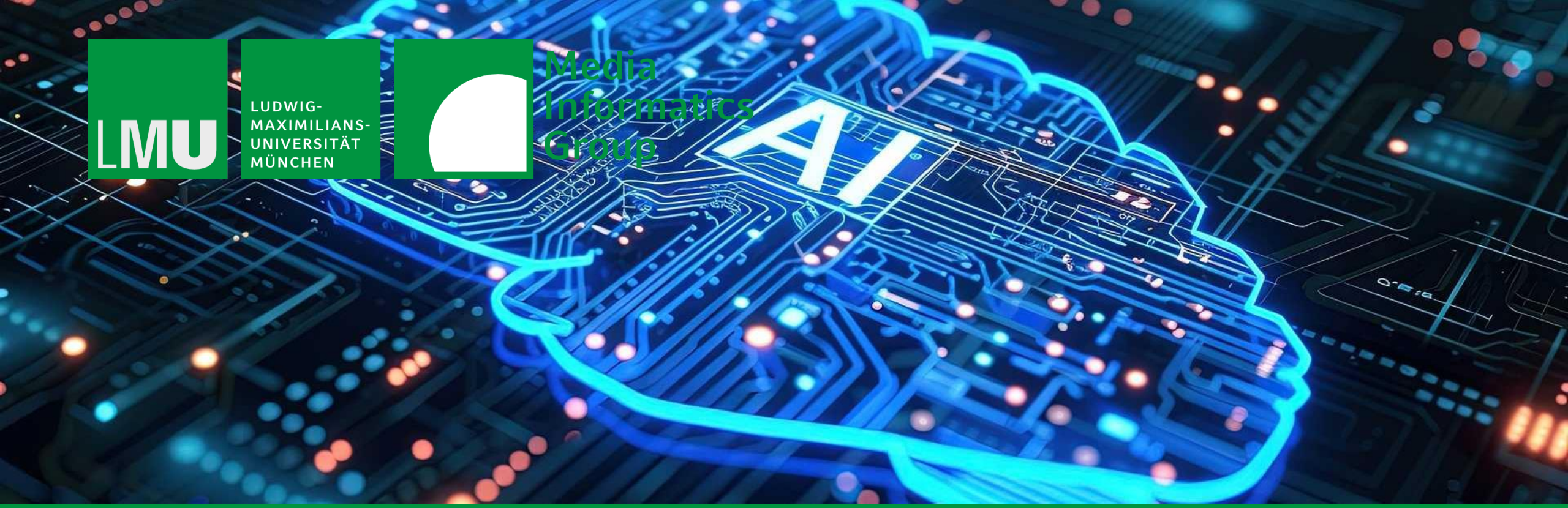



# Local AI Models with LM Studio

Philipp Thalhammer

# Why are local LLMs a Good Idea?

- Keep control of your data
- No (less) censorship
- Works offline
- Cheap
- Great for prototyping

# But…

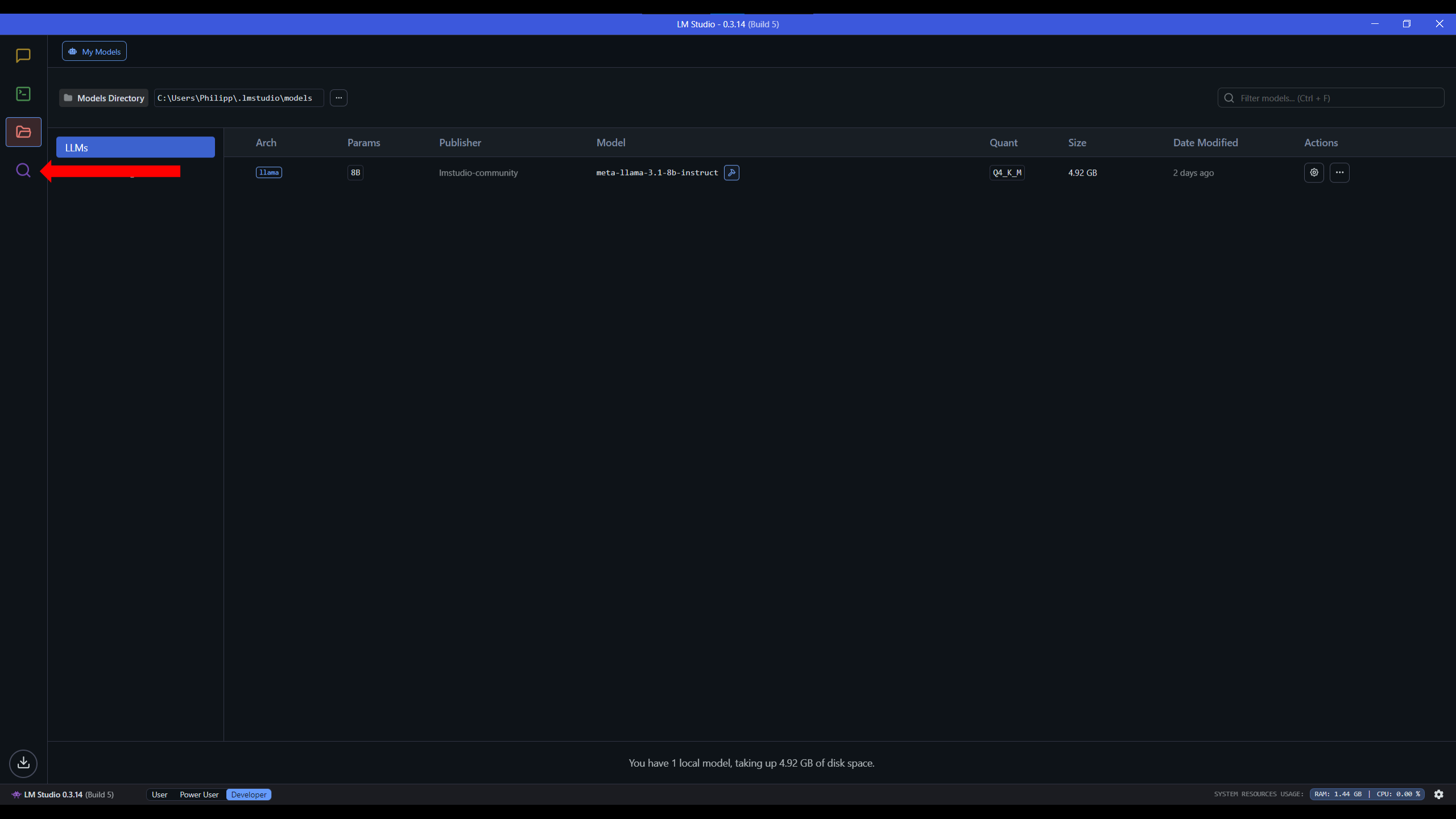
- Not as fast
- Worse performance

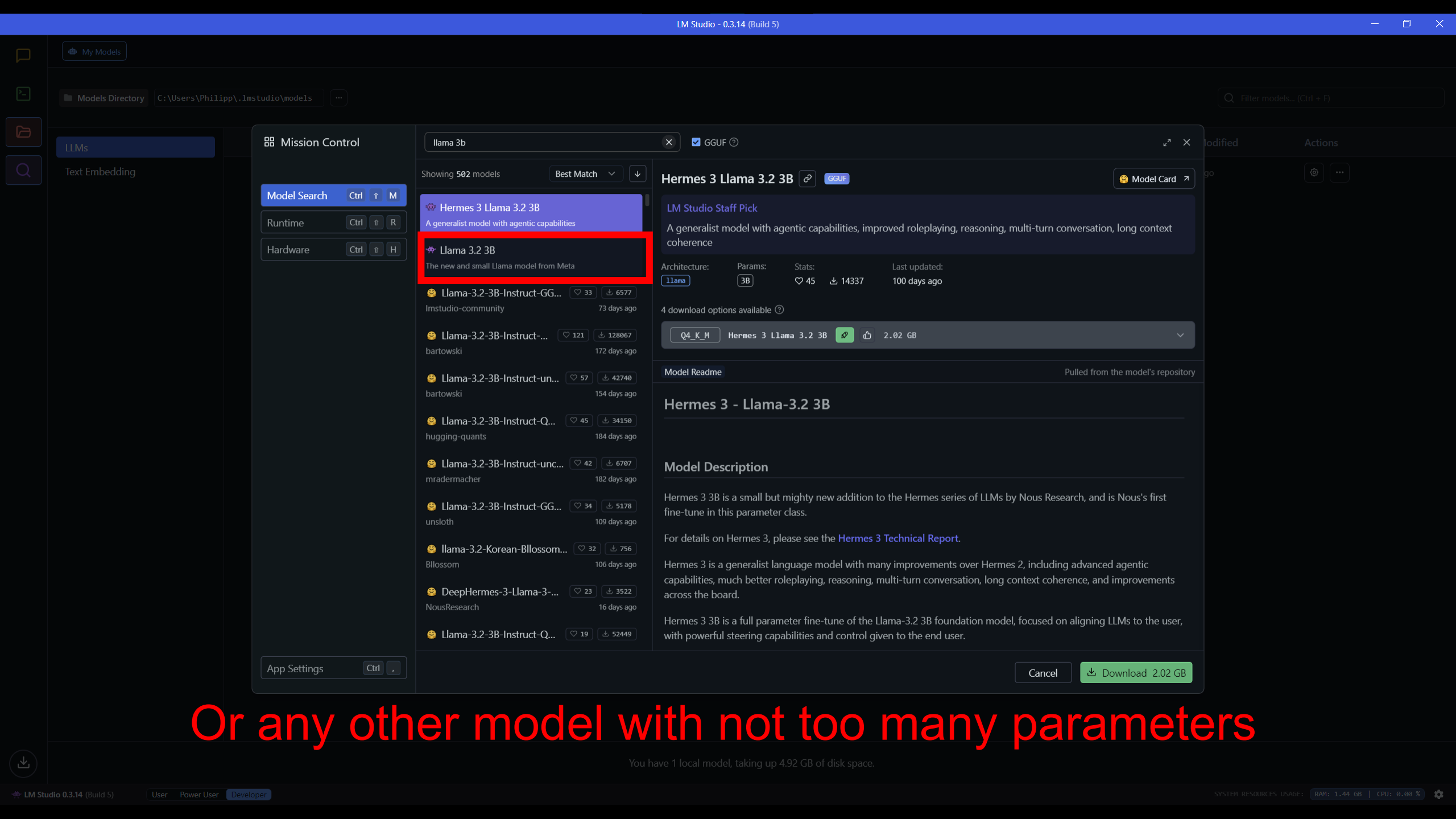
# What You (of Course) Already Did

- Download LM Studio
  - https://lmstudio.ai/

# What You (of Course) Already Did

- Download a *small* model
  - E.G., *meta-llma-3.1-8b-instruct*


LM Studio - 0.3.14 (Build 5)
My Models
Models Directory
C:\Users\Philipp\.lmstudio\models
Filter models... (Ctrl + F)
LLMs
Arch
Params
Publisher
Model
Quant
Size
Date Modified
Actions
llama
8B
lmstudio-community
meta-llama-3.1-8b-instruct
Q4_K_M
4.92 GB
2 days ago
You have 1 local model, taking up 4.92 GB of disk space.
LM Studio 0.3.14 (Build 5)
User
Power User
Developer
SYSTEM RESOURCES USAGE:
RAM: 1.44 GB | CPU: 0.00 %


LM Studio - 0.3.14 (Build 5)
My Models
Models Directory
C:\Users\Philipp\.lmstudio\models
Filter models... (Ctrl + F)
LLMs
Text Embedding
Mission Control
llama 3b
GGUF
Showing 502 models
Best Match
Model Search Ctrl ⇧ M
Runtime Ctrl ⇧ R
Hardware Ctrl ⇧ H
App Settings Ctrl ,
Hermes 3 Llama 3.2 3B
A generalist model with agentic capabilities
Llama 3.2 3B
The new and small Llama model from Meta
Llama-3.2-3B-Instruct-GG... 33 6577
lmstudio-community 73 days ago
Llama-3.2-3B-Instruct-... 121 128067
bartowski 172 days ago
Llama-3.2-3B-Instruct-un... 57 42740
bartowski 154 days ago
Llama-3.2-3B-Instruct-Q... 45 34150
hugging-quants 184 days ago
Llama-3.2-3B-Instruct-unc... 42 6707
mradermacher 182 days ago
Llama-3.2-3B-Instruct-GG... 34 5178
unsloth 109 days ago
llama-3.2-Korean-Bllossom... 32 756
Bllossom 106 days ago
DeepHermes-3-Llama-3-... 23 3522
NousResearch 16 days ago
Llama-3.2-3B-Instruct-Q... 19 52449
Hermes 3 Llama 3.2 3B
GGUF
Model Card
LM Studio Staff Pick
A generalist model with agentic capabilities, improved roleplaying, reasoning, multi-turn conversation, long context coherence
Architecture: llama
Params: 3B
Stats: 45 14337
Last updated: 100 days ago
4 download options available
Q4_K_M Hermes 3 Llama 3.2 3B 2.02 GB
Model Readme
Pulled from the model's repository
Hermes 3 - Llama-3.2 3B
Model Description
Hermes 3 3B is a small but mighty new addition to the Hermes series of LLMs by Nous Research, and is Nous's first fine-tune in this parameter class.
For details on Hermes 3, please see the Hermes 3 Technical Report.
Hermes 3 is a generalist language model with many improvements over Hermes 2, including advanced agentic capabilities, much better roleplaying, reasoning, multi-turn conversation, long context coherence, and improvements across the board.
Hermes 3 3B is a full parameter fine-tune of the Llama-3.2 3B foundation model, focused on aligning LLMs to the user, with powerful steering capabilities and control given to the end user.
Cancel
Download 2.02 GB
Or any other model with not too many parameters
You have 1 local model, taking up 4.92 GB of disk space.
LM Studio 0.3.14 (Build 5)
User
Power User
Developer
SYSTEM RESOURCES USAGE:
RAM: 1.44 GB | CPU: 0.00 %

3. **User story development**: Based on the gathered requirements, I can assist with crafting user stories that describe specific scenarios or use cases for the product or system being prototyped.
4. **Wireframing and concept exploration**: Through text-based descriptions or even simple diagrams, I can help explore different design concepts, wireframes, and user interfaces to identify potential issues early on.
5. **Prototyping and testing**: By generating sample data, test cases, or even prototype code snippets (in certain programming languages), I can aid in the development of a minimum viable product (MVP) for testing and iteration.
6. **Analysis and feedback**: Throughout the prototyping process, I can offer feedback on design decisions, provide insights into potential pitfalls or areas for improvement, and help refine the product based on

# Now What?

ng to achieve with

s to explore ideas

n assist with simple

st assumptions,

ons.

e shared document editing and commenting.

3. **Code editors with chat integration**: Tools like Visual Studio Code, IntelliJ IDEA, or Sublime Text offer built-in chat functionality for seamless collaboration.

By combining these tools and best practices, you can effectively leverage me as a Large Language Model to support your prototyping needs.

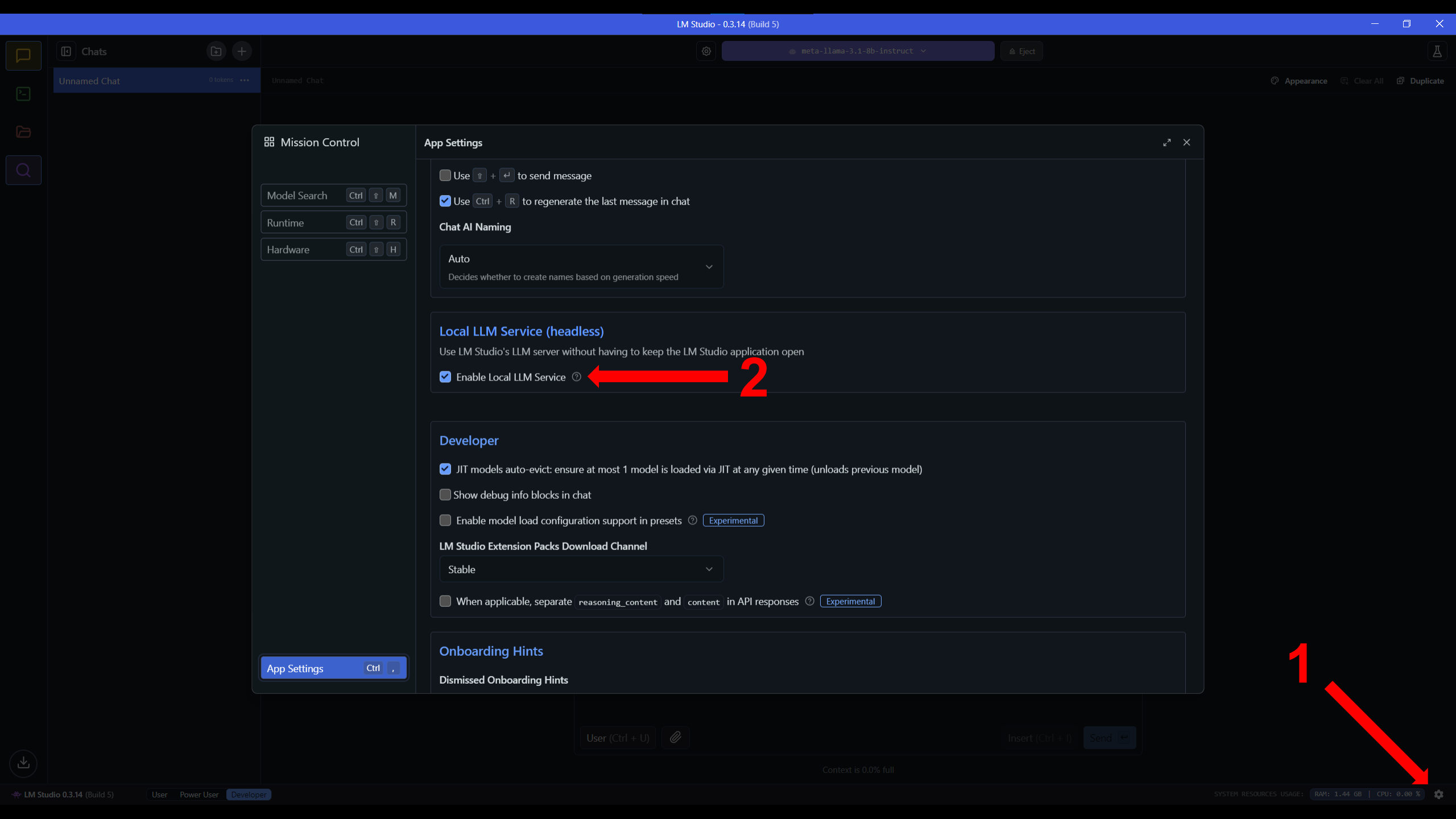
Type a message and press Enter to send ...

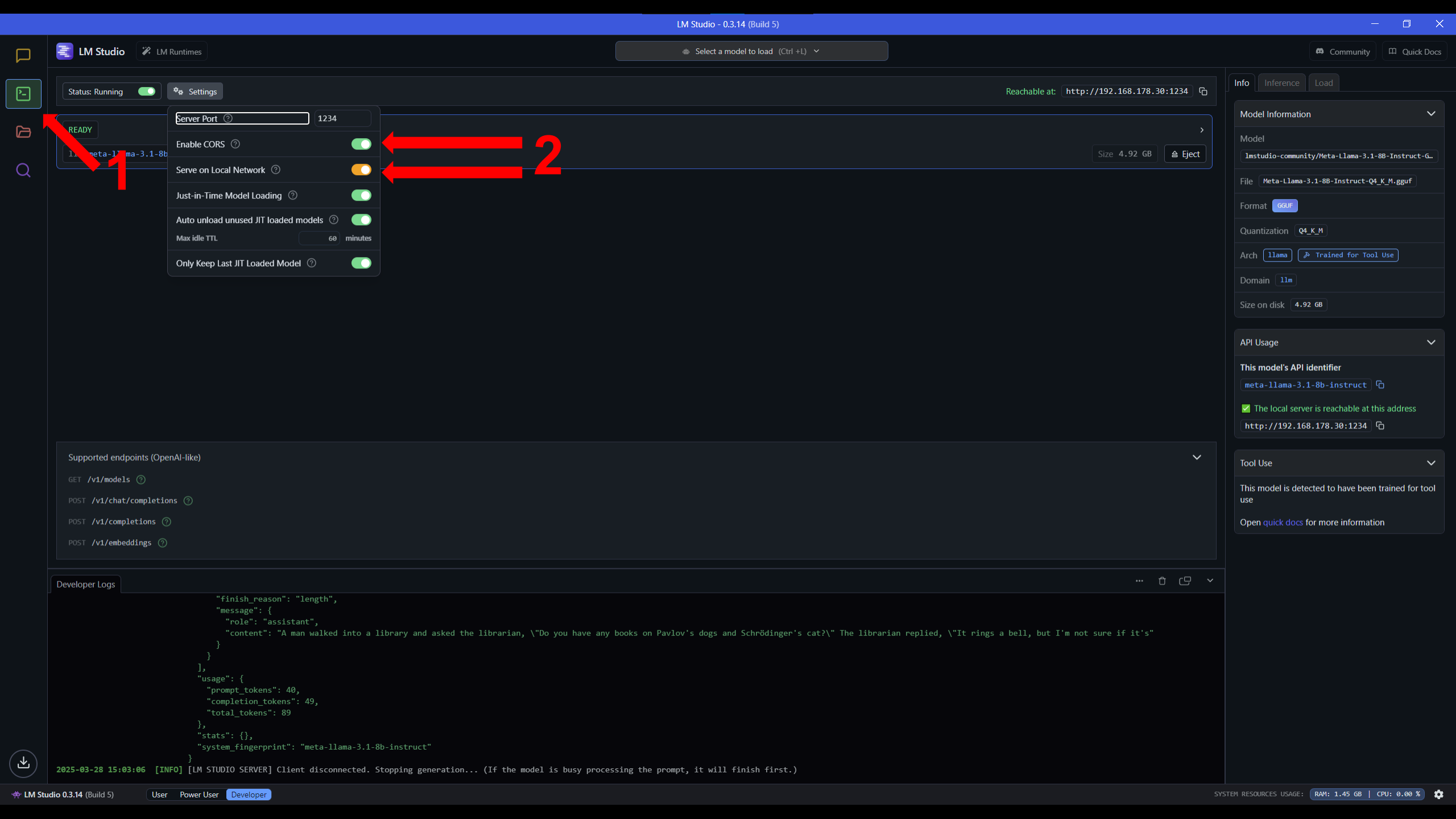
User (Ctrl + U)

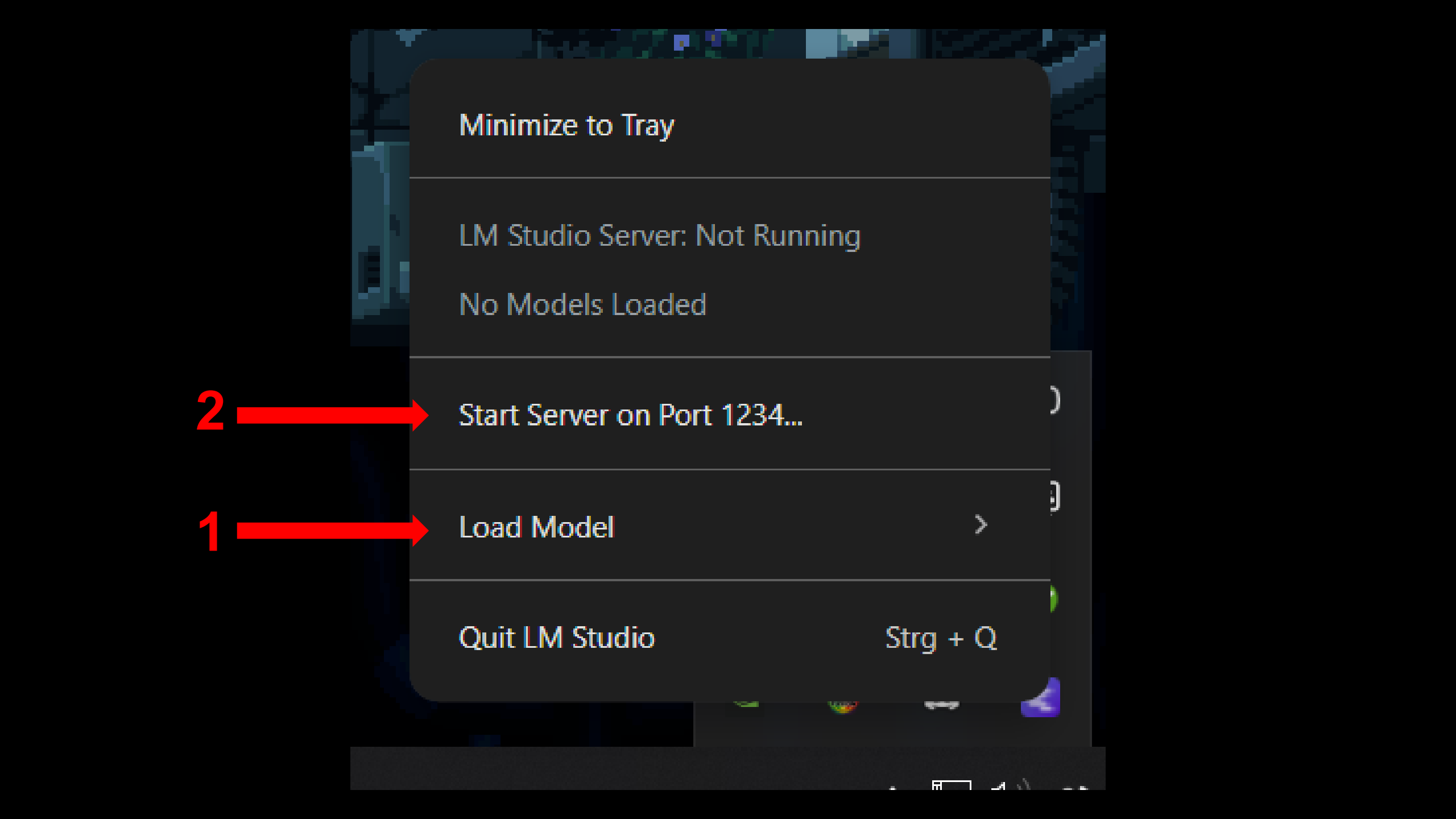
Insert (Ctrl + I) Send ↵

Context is 23.6% full

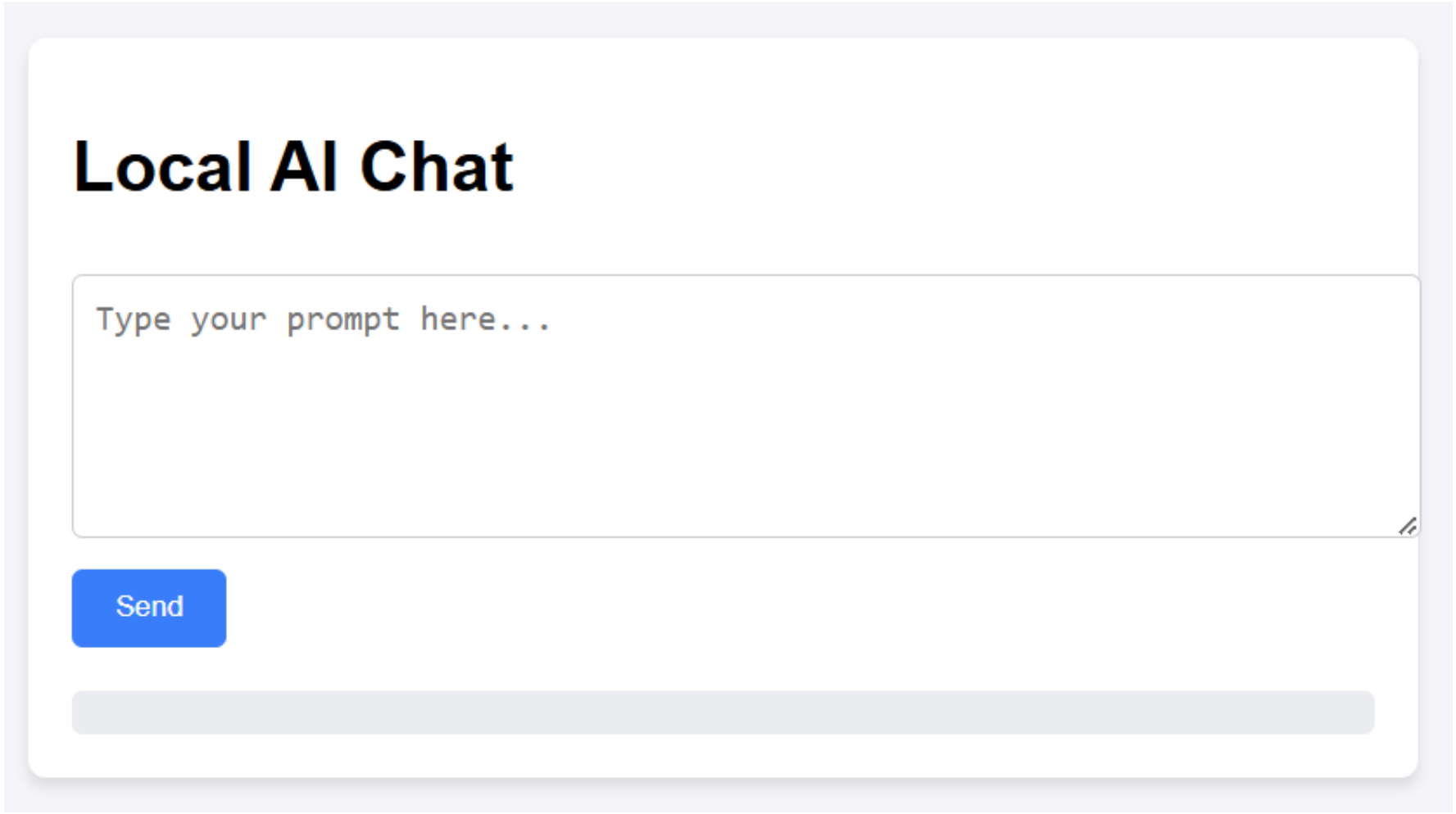
# Let’s set up a server!


LM Studio - 0.3.14 (Build 5)
Chats
Unnamed Chat
0 tokens
meta-llama-3.1-8b-instruct
Eject
Appearance
Clear All
Duplicate
Mission Control
Model Search Ctrl ⇧ M
Runtime Ctrl ⇧ R
Hardware Ctrl ⇧ H
App Settings Ctrl ,
App Settings
Use ⇧ + ↵ to send message
Use Ctrl + R to regenerate the last message in chat
Chat AI Naming
Auto
Decides whether to create names based on generation speed
Local LLM Service (headless)
Use LM Studio's LLM server without having to keep the LM Studio application open
Enable Local LLM Service
2
Developer
JIT models auto-evict: ensure at most 1 model is loaded via JIT at any given time (unloads previous model)
Show debug info blocks in chat
Enable model load configuration support in presets
Experimental
LM Studio Extension Packs Download Channel
Stable
When applicable, separate reasoning_content and content in API responses
Experimental
Onboarding Hints
Dismissed Onboarding Hints
1
User (Ctrl + U)
Insert (Ctrl + I)
Send
Context is 0.0% full
LM Studio 0.3.14 (Build 5)
User
Power User
Developer
SYSTEM RESOURCES USAGE:
RAM: 1.44 GB | CPU: 0.00 %


LM Studio - 0.3.14 (Build 5)
LM Studio
LM Runtimes
Select a model to load (Ctrl +L)
Community
Quick Docs
Status: Running
Settings
Reachable at: http://192.168.178.30:1234
Server Port 1234
Enable CORS
Serve on Local Network
Just-in-Time Model Loading
Auto unload unused JIT loaded models
Max idle TTL 60 minutes
Only Keep Last JIT Loaded Model
READY
Size 4.92 GB
Eject
1
2
Info
Inference
Load
Model Information
Model
lmstudio-community/Meta-Llama-3.1-8B-Instruct-G…
File Meta-Llama-3.1-8B-Instruct-Q4_K_M.gguf
Format GGUF
Quantization Q4_K_M
Arch llama
Trained for Tool Use
Domain llm
Size on disk 4.92 GB
API Usage
This model's API identifier
meta-llama-3.1-8b-instruct
The local server is reachable at this address
http://192.168.178.30:1234
Tool Use
This model is detected to have been trained for tool use
Open quick docs for more information
Supported endpoints (OpenAI-like)
GET /v1/models
POST /v1/chat/completions
POST /v1/completions
POST /v1/embeddings
Developer Logs
"finish_reason": "length",
"message": {
"role": "assistant",
"content": "A man walked into a library and asked the librarian, \"Do you have any books on Pavlov's dogs and Schrödinger's cat?\" The librarian replied, \"It rings a bell, but I'm not sure if it's"
"usage": {
"prompt_tokens": 40,
"completion_tokens": 49,
"total_tokens": 89
"stats": {},
"system_fingerprint": "meta-llama-3.1-8b-instruct"
2025-03-28 15:03:06 [INFO] [LM STUDIO SERVER] Client disconnected. Stopping generation... (If the model is busy processing the prompt, it will finish first.)
LM Studio 0.3.14 (Build 5)
User
Power User
Developer
SYSTEM RESOURCES USAGE:
RAM: 1.45 GB | CPU: 0.00 %


Minimize to Tray
LM Studio Server: Not Running
No Models Loaded
2
Start Server on Port 1234...
1
Load Model
Quit LM Studio
Strg + Q

# HTML Example




Why Does It Talk
Like That?

File Edit Selection View Go Run Terminal Help

LLM Studio Test

EXPLORER

LLM STUDIO TEST

micropython.py 7

webinterface.html

Welcome | webinterface.html | micropython.py 7

webinterface.html > html > head > style > button

```
<html lang="en">
<head>
    <style>
        }
    </style>
</head>
<body>
    <div class="container">
        <h1>Local AI Chat</h1>
        <textarea id="userInput" placeholder="Type your prompt here..."></
        textarea>
        <button onclick="sendRequest()">Send</button>
        <div id="response" class="response"></div>
    </div>

    <script>
    async function sendRequest() {
        const userInput = document.getElementById('userInput').value;
        const responseDiv = document.getElementById('response');

        responseDiv.textContent = 'Processing...';

        try {
            const response = await fetch('http://localhost:1234/v1/chat/
            completions', {
                method: 'POST',
                headers: { 'Content-Type': 'application/json' },
                body: JSON.stringify({
                    model: "local-model",
                    messages: [{ role: "user", content: userInput }
                        { role: "system", content: "Only answer in
                        rhymes" }],
                    max_tokens: 200
                })
            });

            if (!response.ok) {
                throw new Error(`HTTP error! status: ${response.status}`);
            }

            const data = await response.json();
            responseDiv.textContent = data.choices[0].message.content.trim();
        } catch (error) {
            responseDiv.textContent = `Error: ${error.message}`;
        }
    }
    </script>
</body>
</html>
```

OUTLINE

TIMELINE

SVN

VS CODE PETS

3 4 12 Live Share

UTF-8 HTML Go Live

# Now, let's try it with a microcontroller!

LM Studio | LM Runtimes | Select a model to load (Ctrl +L) | Community | Quick Docs

Status: Running | Settings | Reachable at: http://192.168.178.30:1234

READY

llm meta-llama-3.1-8b-instruct | cURL | Size 4.92 GB | Eject

Supported endpoints (OpenAI-like)

GET /v1/models

POST /v1/chat/completions

POST /v1/completions

POST /v1/embeddings

Developer Logs

```
                    "system_fingerprint": "meta-llama-3.1-8b-instruct"
                }
2025-03-31 13:27:56  [INFO] [LM STUDIO SERVER] Client disconnected. Stopping generation... (If the model is busy processing the prompt, it will finish first.)
2025-03-31 14:13:36  [INFO] Server stopped.
2025-03-31 14:42:24  [INFO] [LM STUDIO SERVER] Success! HTTP server listening on port 1234
2025-03-31 14:42:24  [WARN] [LM STUDIO SERVER] Server accepting connections from the local network. Only use this if you know what you are doing!
2025-03-31 14:42:24  [INFO]
2025-03-31 14:42:24  [INFO] [LM STUDIO SERVER] Supported endpoints:
2025-03-31 14:42:24  [INFO] [LM STUDIO SERVER] -> GET http://192.168.178.30:1234/v1/models
2025-03-31 14:42:24  [INFO] [LM STUDIO SERVER] -> POST http://192.168.178.30:1234/v1/chat/completions
2025-03-31 14:42:24  [INFO] [LM STUDIO SERVER] -> POST http://192.168.178.30:1234/v1/completions
2025-03-31 14:42:24  [INFO] [LM STUDIO SERVER] -> POST http://192.168.178.30:1234/v1/embeddings
2025-03-31 14:42:24  [INFO]
2025-03-31 14:42:24  [INFO] [LM STUDIO SERVER] Logs are saved into C:\Users\Philipp\.lmstudio\server-logs
2025-03-31 14:42:24  [INFO] Server started.
2025-03-31 14:42:24  [INFO] Just-in-time model loading active.
```

Info | Inference | Load

Model Information

Model: lmstudio-community/Meta-Llama-3.1-8B-Instruct-G…

File: Meta-Llama-3.1-8B-Instruct-Q4_K_M.gguf

Format: GGUF

Quantization: Q4_K_M

Arch: llama | Trained for Tool Use

Domain: llm

Size on disk: 4.92 GB

API Usage

This model's API identifier

meta-llama-3.1-8b-instruct

The local server is reachable at this address

http://192.168.178.30:1234

Tool Use

This model is detected to have been trained for tool use

Open quick docs for more information

Thonny - MicroPython device :: /main.py @ 39 : 1

File Edit View Run Tools Help

Files

This computer
F: \ Prototyping \
RFID-RC522-Micropython-Driver

- Housing
- create.py
- erase.py
- mfrc522.py
- read.py
- README.md
- rfidaccess.py

MicroPython device

- boot.py
- main.py

[ main.py ] *

```python
import network
import urequests
import ujson
import time

# WiFi credentials
SSID = ""
PASSWORD = ""

def connect_wifi():
    wlan = network.WLAN(network.STA_IF)
    wlan.active(True)
    wlan.connect(SSID, PASSWORD)
    while not wlan.isconnected():
        time.sleep(1)
    print("Connected to WiFi:", wlan.ifconfig())

def send_request():
    url = "http://X.X.X.X:1234/v1/chat/completions"  # Replace with your PC's local IP
    headers = {
        "Content-Type": "application/json"
    }
    data = {
        "model": "your_model_name",  # Replace with the actual model name
        "messages": [{"role": "user", "content": "Tell me a joke!"}],
        "max_tokens": 50
    }

    try:
        print("Sending request to:", url)
        response = urequests.post(url, headers=headers, data=ujson.dumps(data))
        json_data = response.json()  # Convert response to a Python dictionary
        response.close()  # Always close the response to free memory

        # Extract assistant response
        if "choices" in json_data and len(json_data["choices"]) > 0:
            ai_response = json_data["choices"][0]["message"]["content"]
            print("AI Response:", ai_response)
        else:
            print("Error: No response from model")

    except Exception as e:
        print("Request failed:", e)

connect_wifi()
send_request()
```

MicroPython (ESP32) • CP2102 USB to UART Bridge Controller @ COM8

# Resources

https://github.com/Phlipinator/Local-LLM-Examples

# Conclusion

## Questions?

**Contact**

Philipp Thalhammer
LMU Munich
philipp.thalhammer@ifi.lmu.de