Lab 08

Create Agentic AI with Chainlit

Lab Objectives

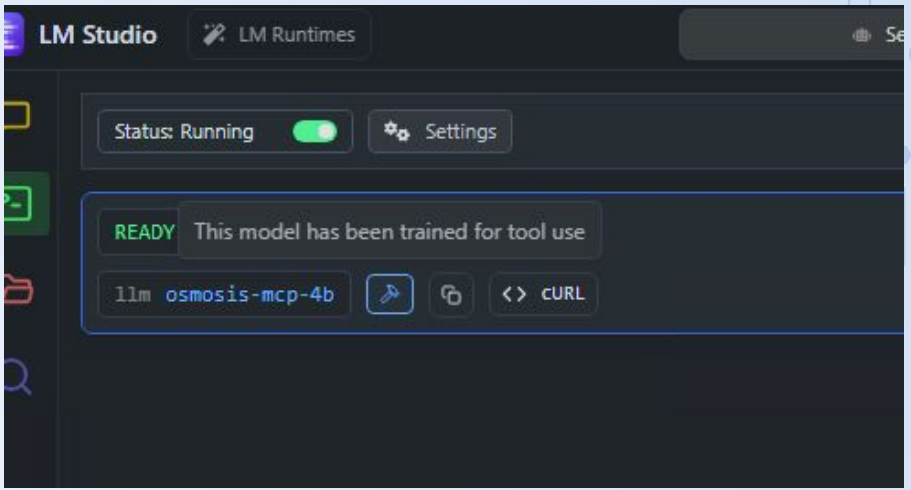
- Learn how Chainlit integrates with LMStudio
- Create an agent to perform an automated task

Create Agentic AI with Chainlit

- Install Python if not already installed
- https://www.python.org/downloads/
- Make sure Python is added to your path variable or know where Python is installed for later steps

Installing a tool capable LLM

- We are going to select a **tool** capable LLM
- You can see which models are capable of executing commands when you see a **hammer** next to the model
- Search for and load the model called **osmosis-mcp-4b**
- If you’re feeling adventurous you can search for a 7B model that also been trained to use tools



Install UV Package Manager

- Install according to your Operating System Instructions
- GitHub - astral-sh/uv: An extremely fast Python package and project manager, written in Rust.

Download (git clone) Obsidian MCP Server

- https://github.com/MarkusPfundstein/mcp-obsidian

Download (git clone) Obsidian MCP Server

- Create a .env file and paste in the following command but with **your API key**
- **If this does not work you may need to load a environment variable into Windows**

```
OBSIDIAN_API_KEY=38a5a5ffe32ce352a36f3354861c827065e72371f1d20c4200062a83264a0c45
```

Download (git clone) Obsidian MCP Server

- Change directory to the root of mcp-obsidian and run the following commands

```
Uv sync
Cd mcp-obsidian-main\src
Python server.py
```

Create Agentic AI with Chainlit

- Optional but recommended:
- Create a virtual environment and activate

```
# Create virtual environment
python -m venv venv

# Activate virtual environment
# On Windows:
venv\Scripts\activate
# On macOS/Linux:
source venv/bin/activate
```

Create Agentic AI with Chainlit

- Install the dependencies

```
pip install -r requirements.txt
```

Create Agentic AI with Chainlit

- Optional: Modify the file to use your **model** of choice (if you are using the prox model provided then skip this step)
- If you changed the **port** the Local Server LMStudio is running on then change this as well

```
pip install -r requirements.txt
```

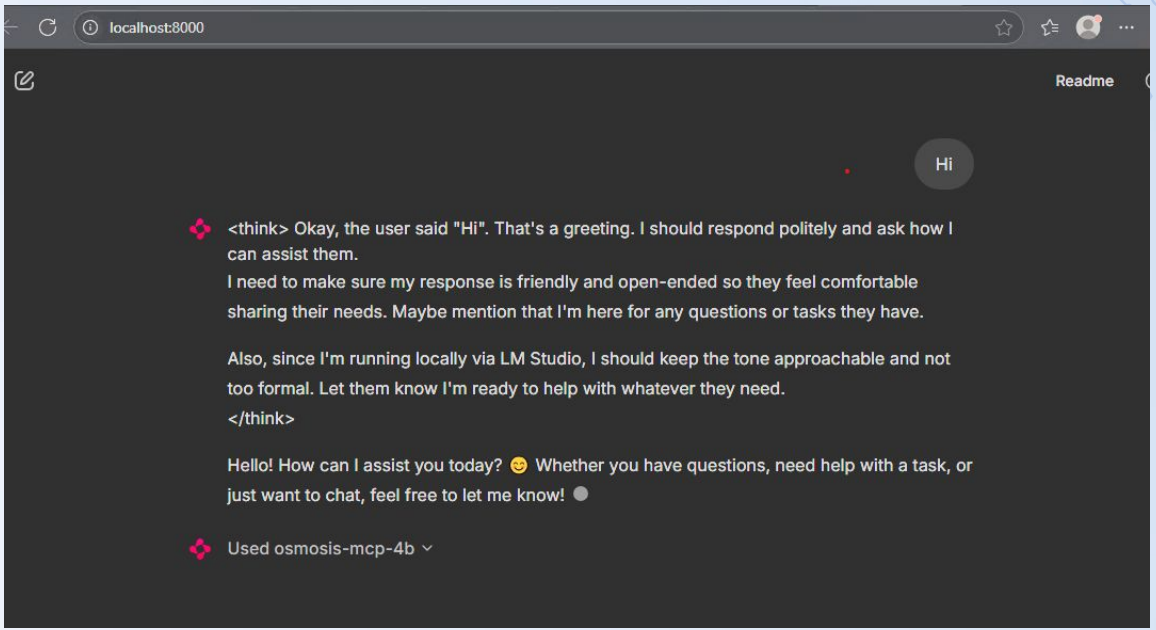
Create Agentic AI with Chainlit

- Run Chainlit

```
chainlit run app.py -w
```

Create Agentic AI with Chainlit

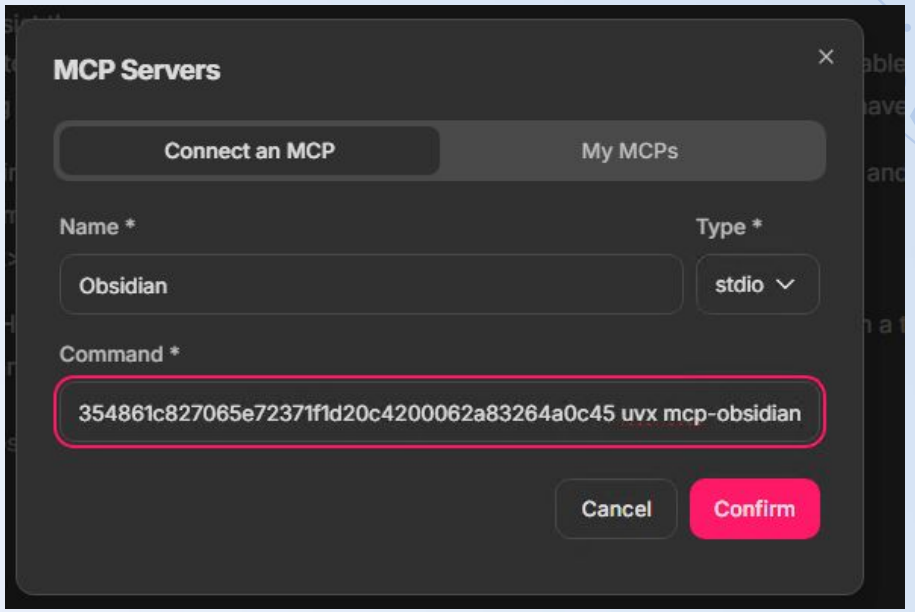
- A browser window will open up
- Test the application by entering Hi. If you receive any errors review the config and if you need help raise your hand!



Create Agentic AI with Chainlit

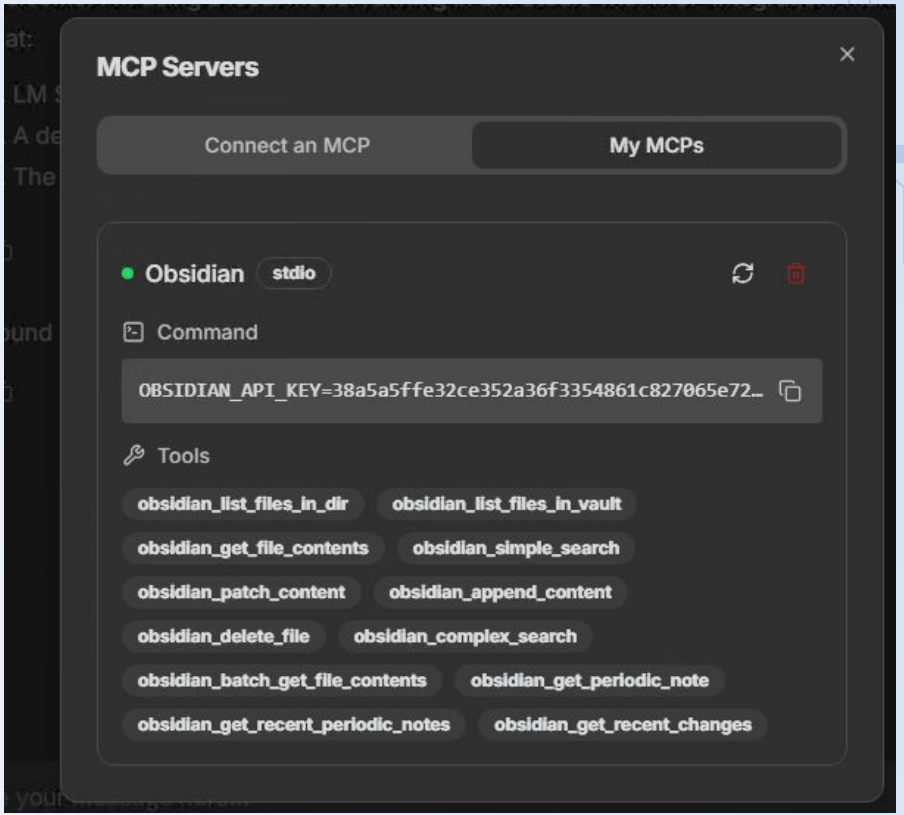
- Next to the chat window there is a plug sign. Select this to add an MCP server and copy the following command
- Make sure to enter your own API key from the Obsidian **Local Rest API** plugin

```
OBSIDIAN_API_KEY=38a5a5ffe32ce352a36f3354861c827065e72371f1d20c4200062a83264a0c45 uvx mcp-obsidian
```



Create Agentic AI with Chainlit

- If the MCP server is successfully added you will see the following success window
- If not, walk through the previous steps
  - If you need help raise your hand!
- You will see all **commands** available to the model under the **tools** section



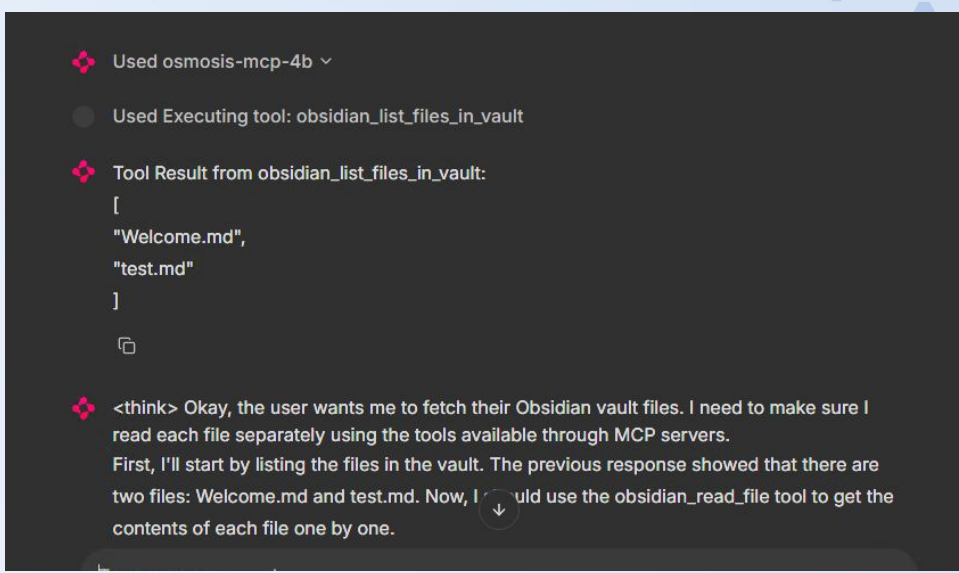
Create Agentic AI with Chainlit

- Type in the following command in Chainlit and execute

Fetch my Obsidian vault files. You must read each of the files with seperate calls carefully and not make up the content yourself. Just make sure you fetch the real file contents via commands first and wait for my instructions. DO NOT repeat the content of the file to me

Create Agentic AI with Chainlit

- If successful you will see it display your note files you have taken during the class!



Create Agentic AI with Chainlit

- Now ask Chainlit to read the files as it requested

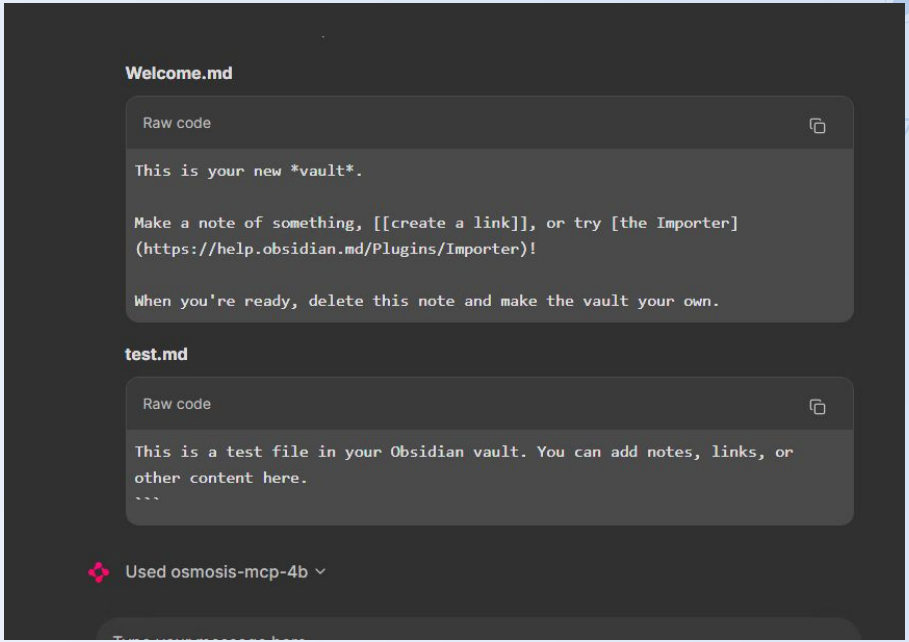
Yes, read these files

Or

Display the text of the files to me

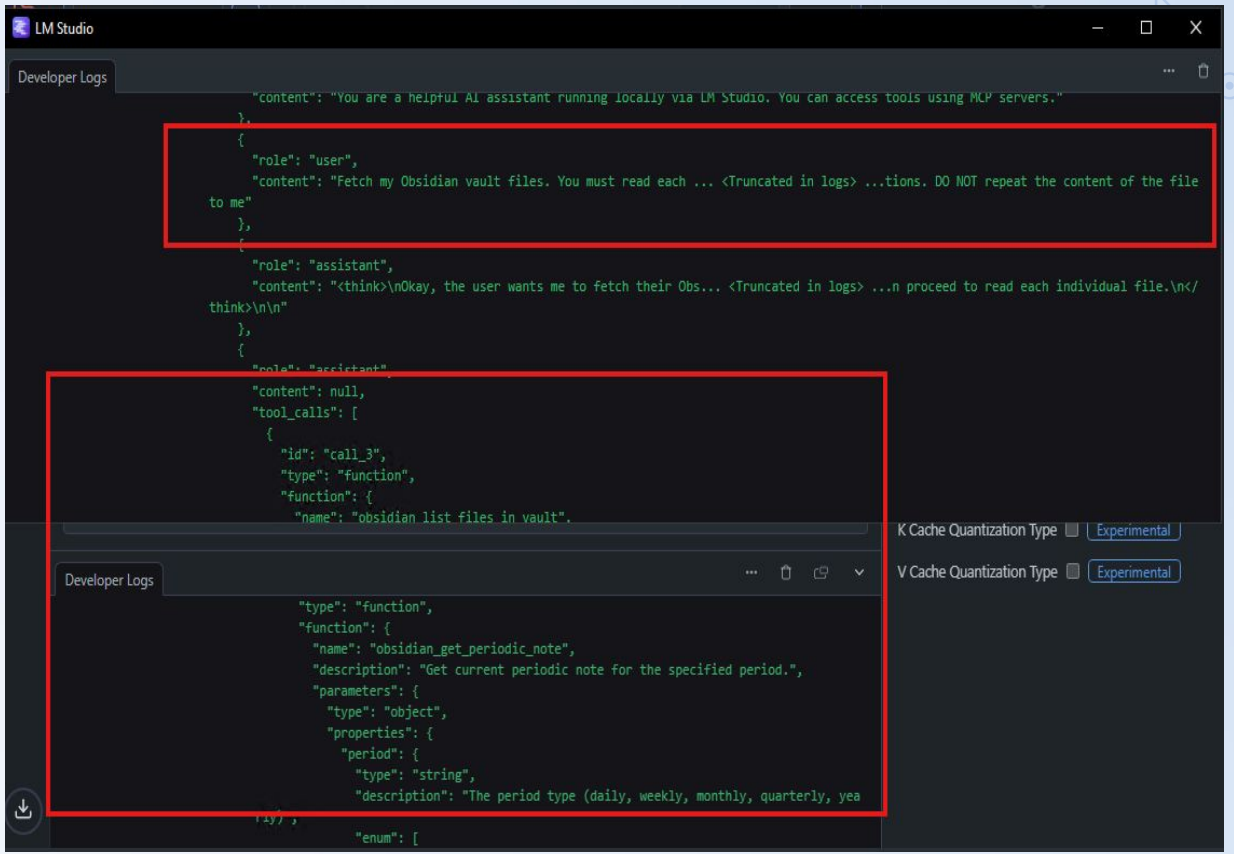
Create Agentic AI with Chainlit

- You should now see your notes you wrote during class



Create Agentic AI with Chainlit

- Open **LMStudio** and review the console logs
- You will see your query to the model
- You will also see the MCP server **injected** all available **functions** and **tools** to the model for selection



Lab End