MCP Server and local files

Overview

In this lesson, we focused on utilizing the **MCP Server** to interact with local files. We explored basic operations such as running the server and using it in conjunction with language models like **Claude** to access and edit local content.

Running the MCP Server

- We first ensured that our server was properly set up and running.
- Remember to use commands like `mcp.run` or `if name == "__main__": mcp.run` to initiate the server.

Testing with Claude

- We tested our setup by integrating it with **Claude**, an LLM (Large Language Model).
- With Claude, we checked the availability of tools that the MCP Server provides.

Interacting with Local Files

- **Read Local Notes:** We asked Claude to access local notes to recall personal preferences, such as favorite pizza toppings.
- **Permission:** For accessing local files, we ensured permission was granted to read from the system.

Adding and Accessing Notes

- We simulated a conversation with Claude, asking about top tennis players, and utilized the MCP Server to add these entries to `notes.txt`.
- Demonstrated updating the notes file with new inputs, showcasing the ability to access past conversations or data in future interactions.

Managing Local Notes

- **Current Method:** Currently, notes are appended to a single file, `notes.txt`.
- **Enhanced Method:** Consider organizing notes by creating a new file for each entry using date-time stamps. This helps in better management and retrieval.

Future Possibilities

- **Search and Indexing:** While our method for reading notes is basic, enhancements like search indexing and vectorization can improve the utility.
- **Expandability:** Notes can be shared and accessed across different MCP clients, allowing memory persistence among various LLMs.

Looking Ahead

- This lesson covered the foundational aspects of MCP Server functionality — reading and writing to local files.
- We touched upon potential applications like automating computer tasks.
- Future modules will delve deeper into building more complex MCP servers and automations.

This concludes our demo on this functionality; onward to exploring more advanced features in subsequent lessons!