Complex inputs with MCP Servers

Overview

In this lesson, we explored more advanced concepts related to **MCP Servers** by diving into the use of complex inputs and custom data schemas. Here's what we covered:

- **Pedantic Inputs**
- **Custom Schemas**
- **Practical Example: Person Class**
- **Integration with MCP**

Pedantic Inputs

- MCP Servers support more than just simple **strings** or **integers**.
- **Pedantic** is a schema and data validation library in Python.
- It allows us to define custom input types, which MCP clients and servers can use effectively.

Custom Schemas

- Custom schemas help by bundling related data into single input objects.
- **Example Schema**: A delivery object might use a **timestamp** and **dimension** as a schema, instead of separate inputs.

Practical Example: Person Class

- We set up a data schema called **person** with attributes:
  - **First Name**: String
  - **Last Name**: String
  - **Years of Experience**: Integer
  - **Previous Addresses**: List of Strings
- This schema makes it easier to manage and manipulate data with multiple attributes.

Integration with MCP

- Created a function `addPersonToMemberDatabase` using the **person** class as input.
- Showcased how to write details from the input into a log file (`log.txt`), simulating a database action.

Highlights

- With a custom schema, functions deal with **bundled complex data** seamlessly.
- **Doc strings** in functions help explain and ensure that input data types are processed correctly by the MCP.

Running the Example

- We demonstrated importing pedantic models into a new Python file, defining schemas, and executing them.

- We saw how the object **person** maps directly to our data and can handle complex types such as lists of strings.

Working with Pedantic

- We interacted with tools like **Claude** to test logging functionality, confirming that the MCP Server correctly handles inputs by resolving number formats and ensuring the correct type.

Through this lesson, we learned how custom data schemas using pedantic can greatly enhance the way we handle inputs in MCP Servers, making our systems more robust and flexible.