Using MCP Servers to interact with your desktop

In this lesson, we explored how MCP servers can automate desktop tasks by running locally on your computer. Let's break down what we covered:

Overview of MCP Server for Desktop Automation

- **Local Benefits**: MCP servers run locally, allowing us to harness the power of Python scripts to interact with desktop resources.
- **Example Task**: We demonstrated how to set up an MCP server to take a screenshot and send it to a client like Claude.

Setting Up Your MCP Server

- **Creating a New File**: We started by creating a new Python file named `screenshot.py`.
- **Setting Up Environment**: Ensured we're in the correct Python environment using `.venv`.
- **Importing Libraries**:
  - **PyAutoGUI**: Essential for taking screenshots.
  - **IO Library**: Used for handling image data.
  - **MCP Specific Imports**: Brought in necessary MCP server utilities and types.

Writing the Screenshot Function

- **Function Purpose**: Created a function to capture and return the current screen as an image.
- **Handling Image Data**: Used a buffer object to hold the screenshot before wrapping it in an image class.

Running the MCP Server

- **Server Execution**: We ran the server with the `MCP.run` command.
- **Connecting and Testing**: Used the MCP inspector to test the server and ensure connectivity. Fixed an error by decorating with `MCP tool`.

Interaction with Claude

- **Configuring Claude**: Ensured the correct path settings in `Claude desktop config.json`.
- **Handling Output**: Tested Claude's ability to process the screenshot and describe its content.
- **Compatibility Check**: Highlighted that not all clients support image outputs, and Claude's compatibility is an advantage.

Key Takeaways

- **Task Automation**: MCP servers can automate desktop tasks using local scripts.
- **Dynamic Outputs**: Showcased that outputs aren't limited to text; they can include images or other media, depending on client compatibility.

- **Client Versatility**: Emphasized the importance of ensuring your client can handle the outputs generated by the server.

This lesson reiterated the power and flexibility of using MCP servers for desktop automation, opening doors to numerous possibilities.