

Lab Handbook: MCP Server Setup and Claude Configuration

Step 1: Prepare the MCP Server Script

Follow the instructions in the provided `.ipynb` file to complete the MCP server code. Export it as a `.py` file and record its absolute path.

Step 2: Configure Claude

1. Open **Claude**.
2. Click the **menu** in the upper-left corner.
3. Navigate to **File** → **Settings**.

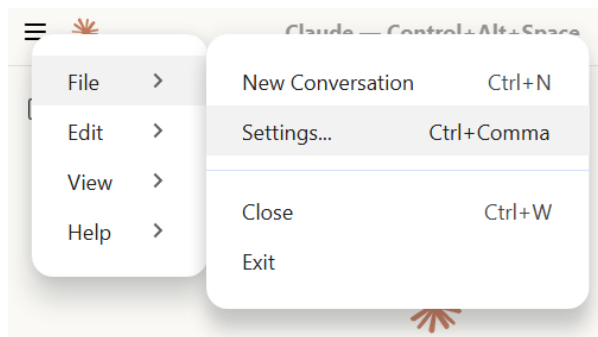


Figure 1: Settings entry point in Claude

4. In the **Settings** panel, select **Developers**.
5. Click **Edit Config**.

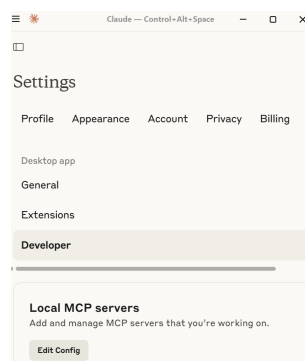


Figure 2: MCP server config entry point

6. Modify the `jsonl` configuration file as follows:

Listing 1: Claude config snippet for MCP server

```
{
  "mcpServers": {
    "<Name of your MCP Server>": {
      "command": "C:\\Path\\to\\python.exe",
      "args": [
        "C:\\Path\\to\\your\\MCPServer.py"
      ]
    }
  }
}
```

Notes:

- <Name of your MCP Server>: Custom MCP server name, e.g., "DHT11".
- `command`: Absolute path to your Python interpreter (e.g., `C:\\Python39\\python.exe` on Windows or `/usr/bin/python3` on Linux/macOS).
- `args[0]`: Absolute path to the MCP server script file (e.g., `D:\\iotmcp\\dht11.py`).
- Use **double backslashes** (\\) for Windows paths, or forward slashes (/) for Linux/macOS paths.
- Ensure the Python environment contains all dependencies required by the MCP server.

Step 3: Start the Toy Communication Protocol

Run the toy communication protocol by executing the corresponding program. In this lab, the IoT system is simulated by a program that returns random sensor data.

Step 4: Test the Setup

You can now interact with the simulated IoT system in natural language through the Claude app.