

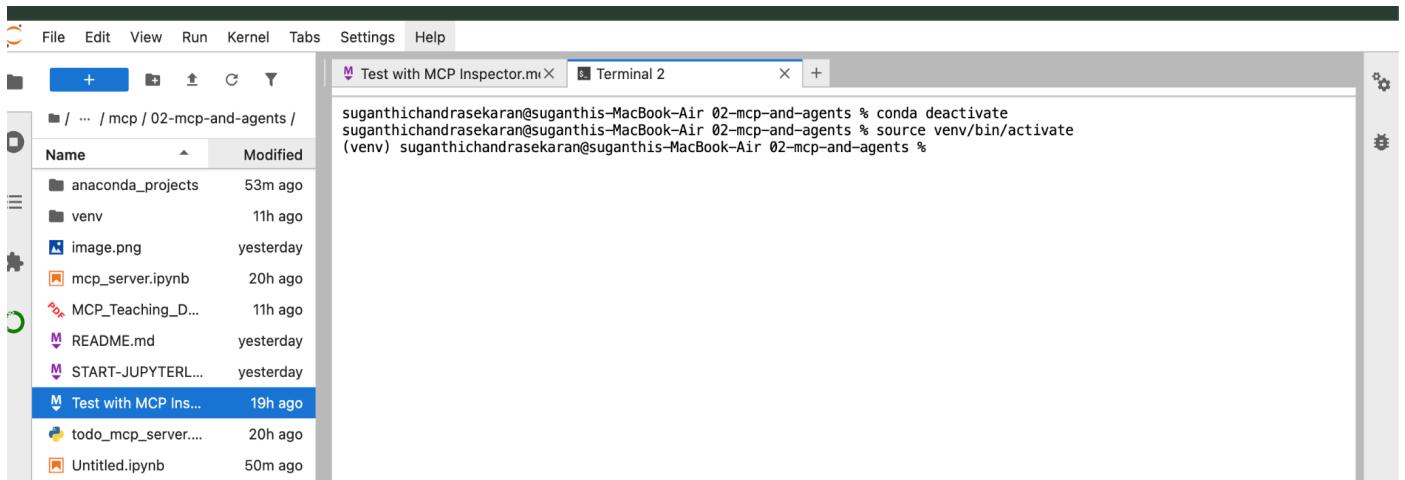
Navigate to the folder , where you have placed your .py file of the mcp server

```
cd
```

```
/Users/suganthichandrasekaran/Documents/AI-engg/new/justmiles/mcp/02-mcp-and-agents
```

```
conda deactivate # run until (base) disappears
```

```
source venv/bin/activate
```



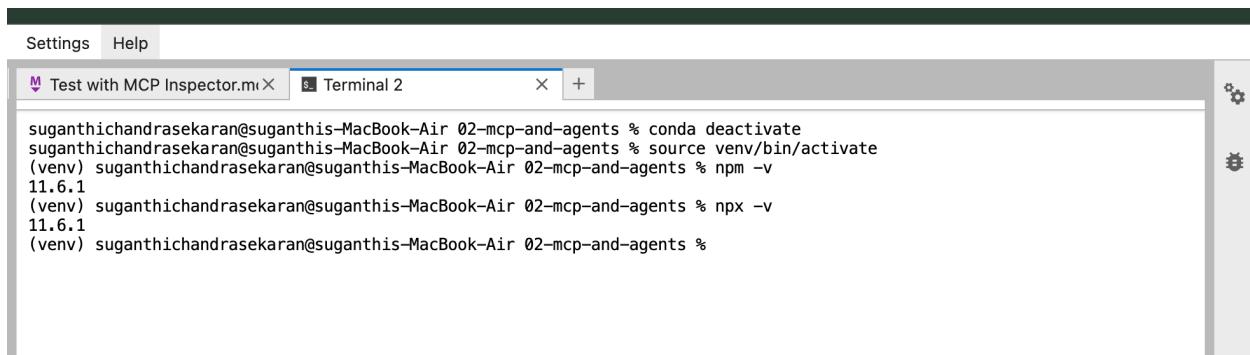
The screenshot shows a Jupyter Notebook interface. On the left is a file browser with the following contents:

- anaconda\_projects (53m ago)
- venv (11h ago)
- image.png (yesterday)
- mcp\_server.ipynb (20h ago)
- MCP\_Teaching\_D... (11h ago)
- README.md (yesterday)
- START-JUPYTERL... (yesterday)
- Test with MCP Ins... (19h ago) - This item is selected.
- todo\_mcp\_server.... (20h ago)
- Untitled.ipynb (50m ago)

The main area shows terminal output:

```
suganthichandrasekaran@suganthis-MacBook-Air 02-mcp-and-agents % conda deactivate
suganthichandrasekaran@suganthis-MacBook-Air 02-mcp-and-agents % source venv/bin/activate
(venv) suganthichandrasekaran@suganthis-MacBook-Air 02-mcp-and-agents %
```

Check if npm is installed . If npm is not available, download and install [node.js](#)



The screenshot shows a Jupyter Notebook interface with a terminal tab open. The output shows the user checking for npm:

```
suganthichandrasekaran@suganthis-MacBook-Air 02-mcp-and-agents % conda deactivate
suganthichandrasekaran@suganthis-MacBook-Air 02-mcp-and-agents % source venv/bin/activate
(venv) suganthichandrasekaran@suganthis-MacBook-Air 02-mcp-and-agents % npm -v
11.6.1
(venv) suganthichandrasekaran@suganthis-MacBook-Air 02-mcp-and-agents % npx -v
11.6.1
(venv) suganthichandrasekaran@suganthis-MacBook-Air 02-mcp-and-agents %
```

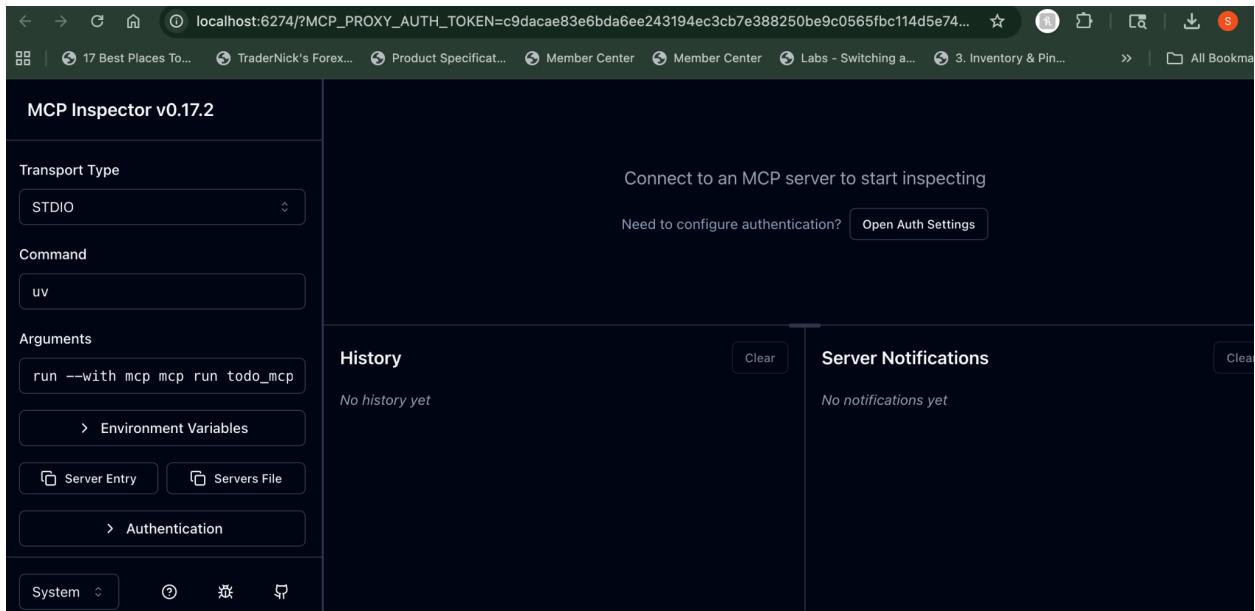
## Check for mcp in your venv

```
suganthichandrasekaran@suganthis-MacBook-Air 02-mcp-and-agents % conda deactivate  
suganthichandrasekaran@suganthis-MacBook-Air 02-mcp-and-agents % source venv/bin/activate  
(venv) suganthichandrasekaran@suganthis-MacBook-Air 02-mcp-and-agents % npm -v  
11.6.1  
(venv) suganthichandrasekaran@suganthis-MacBook-Air 02-mcp-and-agents % npx -v  
11.6.1  
(venv) suganthichandrasekaran@suganthis-MacBook-Air 02-mcp-and-agents % which mcp  
/Users/suganthichandrasekaran/Documents/AI-engg/new/justmiles/mcp/02-mcp-and-agents/venv/bin/mcp  
(venv) suganthichandrasekaran@suganthis-MacBook-Air 02-mcp-and-agents %
```

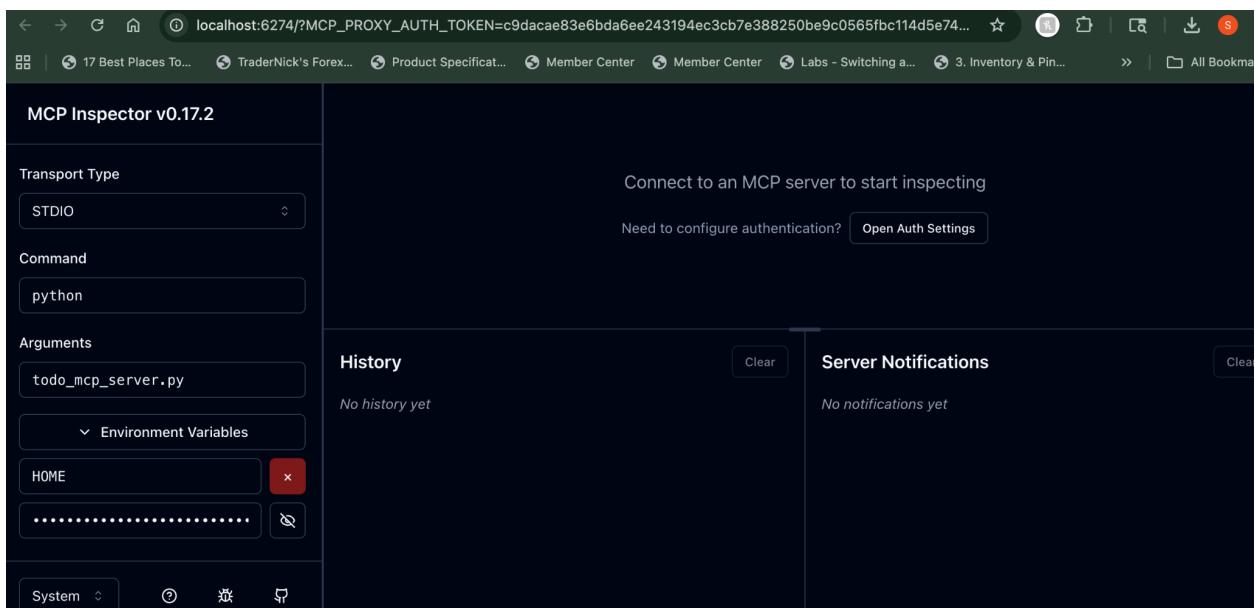
**Invoke MCP Inspector via “mcp dev todo\_mcp\_server.py” → todo\_mcp\_server.py is your mcp server**

```
Test with MCP Inspector.m | Terminal 2 +  
suganthichandrasekaran@suganthis-MacBook-Air 02-mcp-and-agents % conda deactivate  
suganthichandrasekaran@suganthis-MacBook-Air 02-mcp-and-agents % source venv/bin/activate  
(venv) suganthichandrasekaran@suganthis-MacBook-Air 02-mcp-and-agents % npm -v  
11.6.1  
(venv) suganthichandrasekaran@suganthis-MacBook-Air 02-mcp-and-agents % npx -v  
11.6.1  
(venv) suganthichandrasekaran@suganthis-MacBook-Air 02-mcp-and-agents % which mcp  
/Users/suganthichandrasekaran/Documents/AI-engg/new/justmiles/mcp/02-mcp-and-agents/venv/bin/mcp  
(venv) suganthichandrasekaran@suganthis-MacBook-Air 02-mcp-and-agents % mcd dev todo_mcp_server.py  
zsh: command not found: mcd  
(venv) suganthichandrasekaran@suganthis-MacBook-Air 02-mcp-and-agents % mcp dev todo_mcp_server.py  
Starting MCP inspector...  
Proxy server listening on localhost:6277  
Session token: c9dacae83e6bda6ee243194ec3cb7e388250be9c0565fbc114d5e740f5456ac7  
Use this token to authenticate requests or set DANGEROUSLY OMIT_AUTH=true to disable auth  
MCP Inspector is up and running at:  
http://localhost:6274/?MCP_PROXY_AUTH_TOKEN=c9dacae83e6bda6ee243194ec3cb7e388250be9c0565fbc114d5e740f5456ac7  
Opening browser...
```

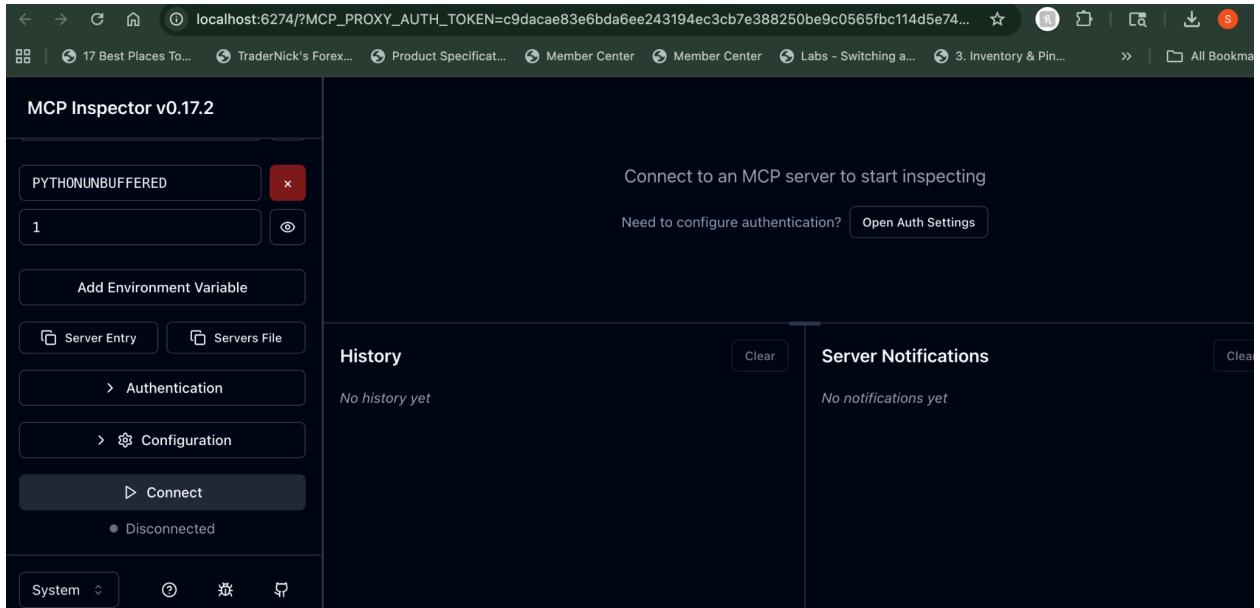
**MCP Inspector will open in port 6274**



**Update command to “python” and Arguments to “todo\_mcp\_server.py”**

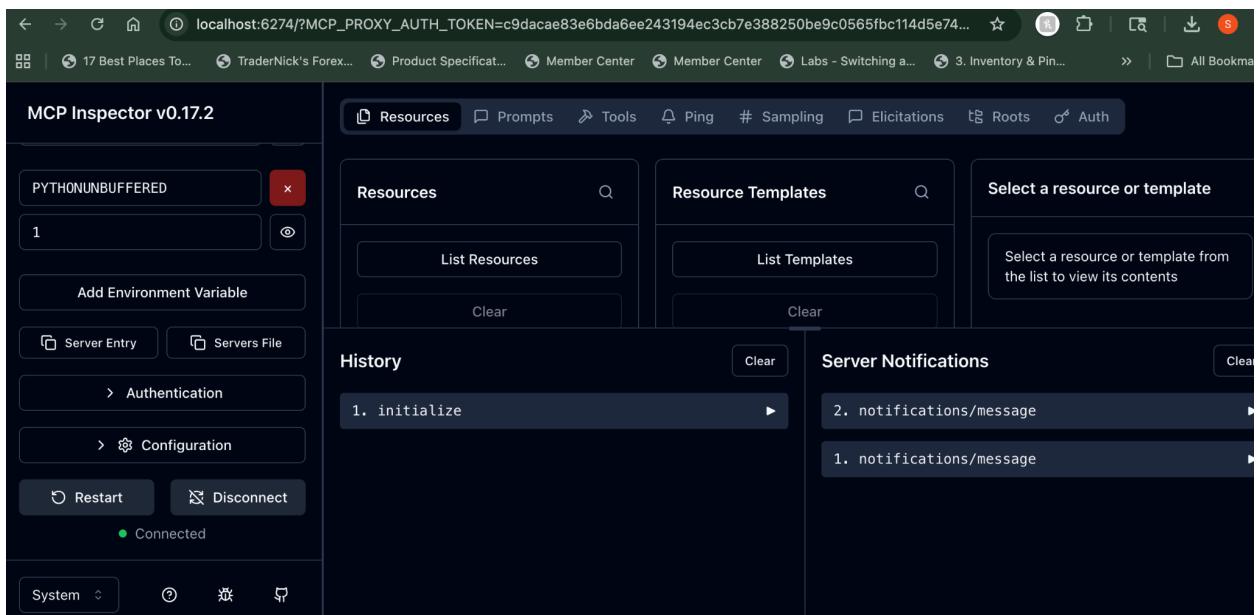


## Add an environment variable PYTHONUNBUFFERED=1



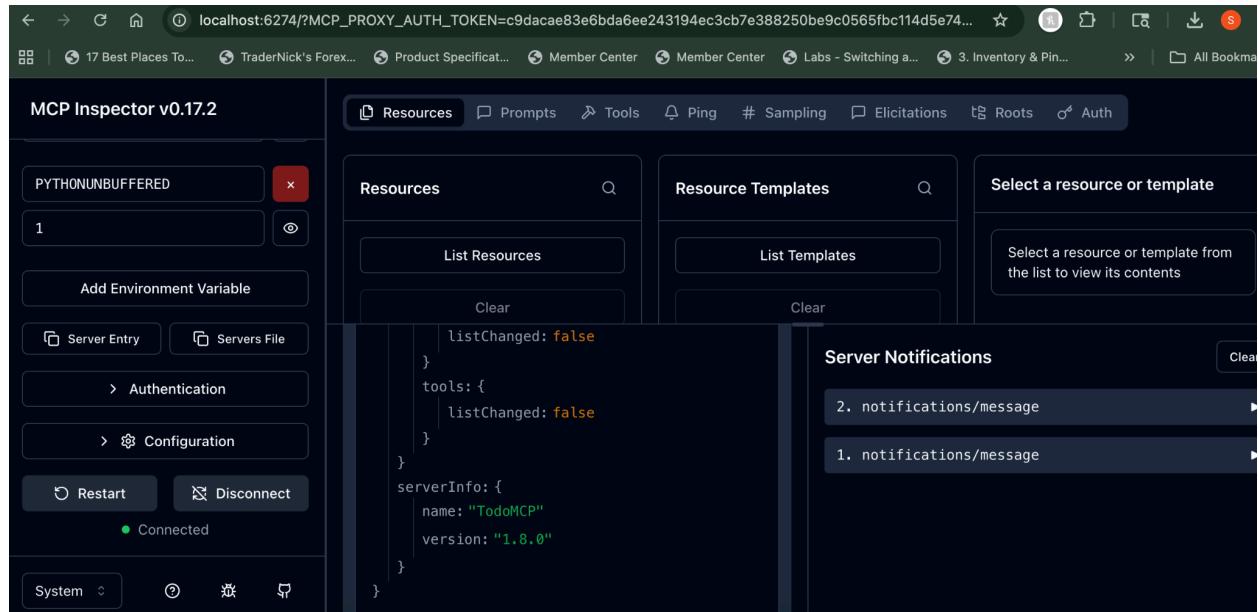
Click on "Connect"

Once the MCP Inspector is connected to your mcp server → It will show as connected.



In the json response of initialize, you will see your mcp server name;

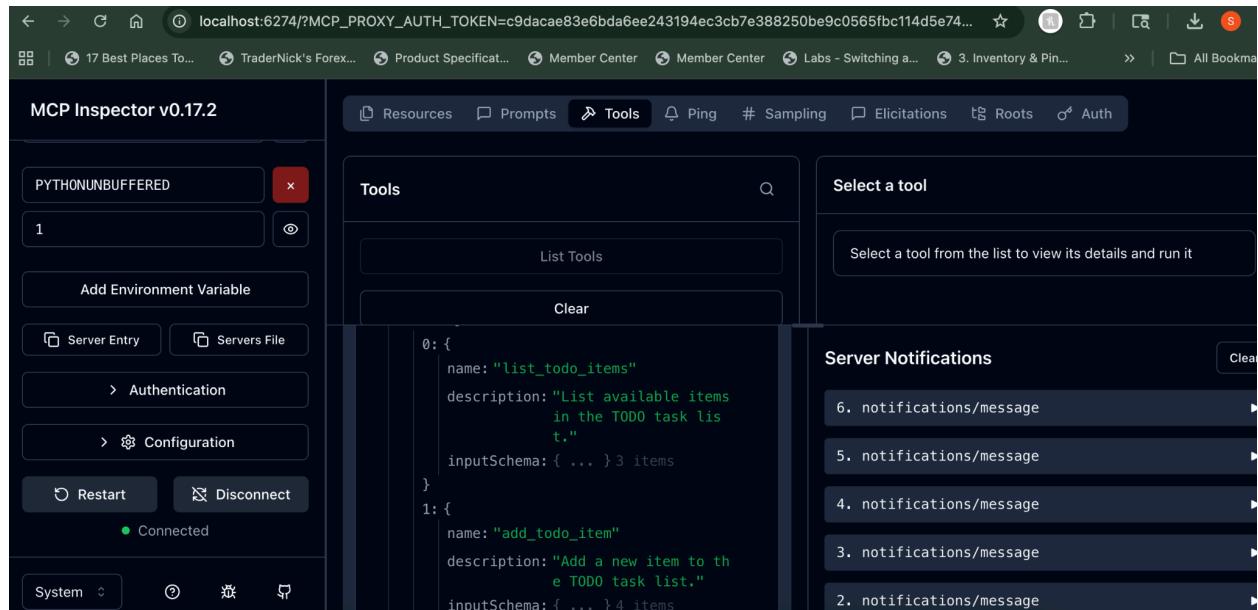
"ToDoMCP" is your mcp server name from the .py file



The screenshot shows the MCP Inspector interface. On the left, there's a sidebar with environment variables, authentication, configuration, and connection status. The main area has tabs for Resources, Prompts, Tools, Ping, Sampling, Elicitations, Roots, and Auth. The Tools tab is active. In the center, under the Tools tab, there's a "Tools" section with a "List Tools" button and a "Clear" button. To the right, there's a "Select a tool" dropdown and a "Server Notifications" section with a "Clear" button. The "Server Notifications" section lists several notifications, including "2. notifications/message" and "1. notifications/message". The central panel displays the JSON response of the initialize call:

```
listChanged: false
}
tools: {
  listChanged: false
}
}
serverInfo: {
  name: "ToDoMCP"
  version: "1.8.0"
}
```

Click on tools in the top menu and 'list tools'. You can see the list of tools in the json response.



The screenshot shows the MCP Inspector interface, similar to the previous one but with the Tools tab selected. The central panel displays the JSON response of the list tools call:

```
0: {
  name: "list_todo_items"
  description: "List available items in the TODO task list."
  inputSchema: { ... } 3 items
}
1: {
  name: "add_todo_item"
  description: "Add a new item to the TODO task list."
  inputSchema: { ... } 4 items
}
```

In a few mins, you will also see the list of tools

The screenshot shows the MCP Inspector interface. On the left, there's a sidebar with various options like 'Add Environment Variable', 'Authentication', and 'Configuration'. Below that are 'Restart' and 'Disconnect' buttons, with a green 'Connected' indicator. The main area has a 'Tools' section with a 'list\_todo\_items' tool selected. This tool has a 'Request' section showing a JSON object:

```
{  
  method: "tools/list",  
  params: {}  
}
```

Below the request is a 'Response' section which is currently empty. To the right of the tool are sections for 'History' (which shows a single entry for the tool) and 'Server Notifications' (which lists four notifications).

Now, select the tool that you wanted to test .

On selecting the tool, you will get the below screen

This screenshot shows the same MCP Inspector interface after the 'list\_todo\_items' tool has been run. The 'Response' section now contains the following JSON output:

```
[  
  {"id": 1, "text": "Test Item 1"},  
  {"id": 2, "text": "Test Item 2"},  
  {"id": 3, "text": "Test Item 3"}]
```

The other sections ('History' and 'Server Notifications') remain the same as in the previous screenshot.

On clicking the “Run tool” you can see the result of “list\_todo\_items”

The screenshot shows the MCP Inspector interface. On the left, there's a sidebar with various tools like PythonUnbuffered, Add Environment Variable, and a section for Authentication and Configuration. The main area is titled "Tools" and contains three items: "List Tools", "Clear", and "list\_todo\_items". Below "list\_todo\_items", it says "List available items in the TODO task list." and shows two items: "1: Create an app" and "2: Deploy an app". To the right, there's a "Tool Result" section with "Success" status and a "History" section showing "1. tools/call". The "Server Notifications" section indicates "No notifications yet".

Selected the tool “add\_todo\_item” and added an item

This screenshot is similar to the previous one but shows the result of running the "add\_todo\_item" tool. The "item \*" field contains "Modify an app". The rest of the interface is identical to the first screenshot, showing the list of todo items and the success message for the tool run.

The screenshot shows the MCP Inspector interface. On the left, there's a sidebar with environment variables like 'PYTHONUNBUFFERED' and 'System'. The main area has a 'Tools' section with 'List Tools' and 'Clear' buttons. A 'list\_todo\_items' tool is listed, described as 'List available items in the TODO task list.' Below it is the 'add\_todo\_item' tool, described as 'Add a new item to the TODO task list.' A 'complete\_todo\_item' tool is also listed, described as 'Mark an item completed by removing it from the list.' To the right, a 'Tool Result: Success' message is shown with the text '"Added 3: Modify an app"'. The 'History' section shows a request for '2. tools/call'. The 'Server Notifications' section shows a notification for '4. notifications/message'.

Now , calling the list tool again

This screenshot is similar to the previous one but shows the 'list\_todo\_items' tool being used. The 'Tool Result: Success' message now displays three items: "'1: Create an app'", "'2: Deploy an app'", and "'3: Modify an app''. The rest of the interface remains the same, with the 'History' section showing '3. tools/call' and the 'Server Notifications' section showing '7. notifications/message'.

Now, called the tool "complete\_todo\_item" and gave the inputs to remove item 2

The screenshot shows the MCP Inspector interface. On the left, there's a sidebar with various tools like 'PYTHONUNBUFFERED', 'Add Environment Variable', and 'Authentication'. The main area has a 'Tools' section with a search bar. Under 'Tools', there are three items: 'list\_todo\_items', 'add\_todo\_item', and 'complete\_todo\_item'. The 'complete\_todo\_item' tool is selected and has an input field 'item\_id \*' containing '2'. Below it are buttons for 'Run Tool' and 'Copy Input'. The 'Tool Result' section shows 'Success' with the message '"Completed 2: Deploy an app"'. At the bottom, there's a 'History' section with '4. tools/call' and a 'Server Notifications' section with '11. notifications/message'.

Calling the first tool , to list the items .

You can see the second item is not present in the list now.

This screenshot shows the same MCP Inspector interface after running the 'complete\_todo\_item' tool. The 'list\_todo\_items' tool is now selected in the 'Tools' section. Its description says 'List available items in the TODO task list.' Below it are buttons for 'Run Tool' and 'Copy Input'. The 'Tool Result' section shows 'Success' with the message '"1: Create an app"'. In the 'History' section, the entry '4. tools/call' is followed by '5. tools/call'. In the 'Server Notifications' section, the entry '11. notifications/message' is followed by '14. notifications/message'.