

A demonstration MCP server is configured, tested, fixed and documented

You practise this in the course

claude_desktop_config.json

```
{
  "mcpServers": {
    "demo-orders": {
      "command": "uv",
      "args": [ "--directory", "/path/to/demo-orders",
                "run", "server.py" ],
      "env": { "DEMO_API_KEY": "set locally, never shared" }
    }
  }
}
```

Test log

Test	Result	Action
Tools listed	list_orders, get_order	–
“List last week’s orders”	401 Unauthorized: API key missing	Key added to the server environment locally
Same query again	12 sample orders returned	Passed

Permissions record

Item	Value
Tools	list_orders, get_order (read-only)
Data	Fictional sample orders
Limitations	No write tools; no customer contact details
Owner	Named IT owner
Review	Quarterly

YOU START WITH

An approved demonstration server with two read-only tools over sample order data.

THE PROMPT USED

“Configure the demo-orders server in the Claude desktop app, confirm its two read-only tools appear, run a test query, and record the server’s permissions, limitations and owner.”

STILL NEEDS HUMAN CHECKING

- Where the API key is stored
- That only read tools are enabled
- The owner and review date with the business

WHAT YOU PRACTISE

Adding a server to the Claude desktop configuration, running a read-only test, diagnosing an access failure and writing the permissions record.

THE FINISHED OUTPUT CONTAINS

- A working configuration with no secrets in shared files
- A test log: success, failure and fix
- A permissions record: tools, data, limits, owner, review date

RELATED COURSE

[Claude MCP: Connect and Configure](#) · Module: Configuration and authentication · Testing and troubleshooting

Account note: In the course you use your own Claude desktop app and a test server you are permitted to run. File locations follow the official MCP documentation.