workato

HOW WE MCP // PART 1 OF 4

Design 101

The fastest way to build a confusing MCP Server is to start with the application. Here's the design order that actually works.

The Shift

What can we expose? --> Who needs it, and to do what?

LLMs operate on: Intent --> Action --> Outcome

Your server should map to this—not to an API schema

When you start with the application...

You end up with

- Multiple unrelated personas in one server
- Competing workflows with no clear priority
- A growing list of "might be useful" tools
- Ambiguous tool names that overlap in scope

The same request maps to **multiple valid tools.** The LLM has to guess which path is correct—and behavior becomes inconsistent. The LLM isn't confused. **The design is.**

Good MCP design removes ambiguity. Bad MCP design creates it.

Design in this order...

1. **Define the persona**

   Who is this for? If the answer is "multiple unrelated roles", that's your first red flag. **One server, one primary user.**

2. **Define the core user cases**

   Not "manage customers", but **"investigate a billing issue" or "update deal stages after a call."** Specific, completable actions.

3. **Define the workflow boundary**

   What does "done" look like? A good server lets the LLM complete a **meaningful unit of work ent-to-end** without bouncing across unrelated concepts.

4. **Now design your tools**

   Once persona, use cases, and workflows are clear, the right tools become obvious. **Tools aren't the starting point. They're the outcome.**

3 questions before you write a tool

- [ ] Can I describe this server in one sentence?
- [ ] Is it clearly for one persona (or related ones)?
- [ ] Can a core use case complete ent-to-end within this server?

If any answer is "no", you don't have a tool problem. You have a scope problem.